Zuzlewski Stéphane January 2000

parametric information schema

Volume I Version 1.5

Table of Contents

1.- Introduction :

This paper is about the design of the NCEDC schema for the parametric information. This schema was designed to be able to store all the information from the CNSS format, following as much as possible the CSS 3.0 schema.

There are 15 relations in the schema. The information stored in these relations includes:

- Event information, including preferred origin, preferred magnitude, etc.
- Multiple origins for an event.
- Multiple magnitudes for each origin.
- Arrivals (seismic signals).
- Association of arrivals with origins.
- Association of amplitudes with origins, as well as with magnitudes.
- Moment tensor information.
- Association of coda information with magnitudes.
- Association of coda information with origins.

The remainder of this volume consists of five chapters:

Chapter 2	*Database structure* Each relation is defined.
Chapter 3	*Database relations* The logical design of the database is expressed in an Entity-Relationship diagram and each relation is described to identify the key fields and the links among the relations.
Chapter 4	*Database attributes* Each attribute is described.
Chapter 5	*Primary keys* The notion of sequence is described.
Chapter 6	*Time issues* A *PL/SQL* implementation of time conversion is given.

<u>2.- Database structure :</u>

This chapter defines the physical structure of each table, as it exists within the DBMS data dictionary. The name of the relation appears in **bold** print at the top left and a description of the table at the top right. **Bold** attributes represent primary keys, *italic* attributes represent foreign keys, and <u>underlined</u> attributes stand for CSS attributes.

Event		Summary information on an event
Attribute	***Type***	***Description***
<u>evid</u>	int	Event identifier
<u>prefor</u>	int	Preferred origin
prefmag	int	Preferred magnitude
prefmec	int	Preferred mechanism
<u>commid</u>	int	Comment identifier
<u>auth</u>	char(15)	Source/originator
subsource	char(8)	Subsystem source
<u>etype</u>	char(7)	Event type
selectflag	int	“Definitive” definition of the event
<u>lddate</u>	date	Load date

Significant_Event		Summary information on a significant event
Attribute	***Type***	***Description***
<u>evid</u>	int	Event identifier
<u>evname</u>	char(80)	Event name
remarks	char(2)	Event indications (felt, damage, ...)
nfelt	int	Number of felt reports
mmi	int	Maximum MM intensity
pga	float	Peak acceleration
<u>lddate</u>	date	Load date

Origin		Data on event location
Attribute	***Type***	***Description***
<u>orid</u>	int	Origin identifier
<u>evid</u>	int	Event identifier
prefmag	int	Preferred magnitude
prefmec	int	Preferred mechanism
<u>commid</u>	int	Comment identifier
bogusflag	int	Bogus origin flag
datetime	float	Origin date
<u>lat</u>	float	Estimated latitude
<u>lon</u>	float	Estimated longitude
<u>depth</u>	float	Estimated depth
type	char(2)	Type of location
<u>algorithm</u>	char(15)	Location algorithm used
algo_assoc	char(80)	Associator algorithm used
<u>auth</u>	char(15)	Source/originator
subsource	char(8)	Subsystem source
datumhor	char(8)	Datum type for latitude & longitude
datumver	char(8)	Datum type for vertical
gap	float	Maximum azimuthal gap of stations
distance	float	Distance to nearest station
wrms	float	Standard error of observations
<u>stime</u>	float	Origin time error
erhor	float	Horizontal error
sdep	float	Depth error
erlat	float	Error in latitude
erlon	float	Error in longitude
totalarr	int	Total number of arrivals
totalamp	int	Total number of amplitude readings
<u>ndef</u>	int	Number of locating phases
nbs	int	Number of S readings
nbfm	int	Number of P first motions
locevid	char(12)	Local event identifier
quality	float	Quality of solution
fdepth	char	Fixed depth
fepi	char	Fixed epicenter
ftime	char	Fixed time
vmodelid	int	Velocity model identifier
cmodelid	int	Crustal model identifier
rflag	char	Human/automatic/finalized observation
<u>lddate</u>	date	Load date

Origin_Error		Error data on event location
Attribute	***Type***	***Description***
orid	int	Origin identifier
sxx	float	Covariance matrix element
syy	float	Covariance matrix element
szz	float	Covariance matrix element
stt	float	Covariance matrix element
sxy	float	Covariance matrix element
sxz	float	Covariance matrix element
syz	float	Covariance matrix element
stx	float	Covariance matrix element
sty	float	Covariance matrix element
stz	float	Covariance matrix element
azismall	float	Azimuth of smallest principal error
dipsmall	float	Dip of smallest principal error
magsmall	float	Magnitude of smallest principal error
aziinter	float	Azimuth of intermediate principal error
dipinter	float	Dip of intermediate principal error
maginter	float	Magnitude of intermediate principal error
azilarge	float	Azimuth of largest principal error
diplarge	float	Dip of largest principal error
maglarge	float	Magnitude of largest principal error
lddate	date	Load date

Netmag		Summary information on a magnitude
Attribute	***Type***	***Description***
magid	int	Magnitude identifier
orid	int	Origin identifier
commid	int	Comment identifier
magnitude	float	Magnitude
magtype	char(6)	Magnitude type
auth	char(15)	Source/originator
subsource	char(8)	Subsystem source
magalgo	char(15)	Magnitude algorithm
nsta	int	Number of stations used
uncertainty	float	Magnitude uncertainty
gap	float	Maximum azimuthal gap of stations
distance	float	Minimum distance of stations
quality	float	Subjective quality
rflag	char	Human/automatic/finalized observation
lddate	date	Load date

Arrival		Summary information on a seismic arrival
Attribute	***Type***	***Description***
arid	int	Arrival identifier
commid	int	Comment identifier
datetime	float	Pick date
sta	char(6)	Station code
net	char(8)	Network identifier
auth	char(15)	Source/originator
subsource	char(8)	Subsystem source
channel	char(3)	Channel name
channelsrc	char(8)	Domain of channel name
seedchan	char(3)	SEED channel name
location	char(2)	SEED location
iphase	char(8)	Reported phase
qual	char	Signal onset quality
clockqual	char	Clock quality
clockcorr	int	Clock correction
ccset	char	Clock correction set
fm	char(2)	First motion
ema	float	Emergence angle
azimuth	float	Observed azimuth
slow	float	Observed slowness
deltim	float	Delta time
delinc	float	Error in angle
delaz	float	Delta azimuth
delslo	float	Delta slowness
quality	float	Quality
snr	float	Signal to noise ratio
rflag	char	Human/automatic/finalized observation
lddate	date	Load date

AssocArO		Data associating arrivals with origins
Attribute	***Type***	***Description***
orid	int	Origin identifier
arid	int	Arrival identifier
commid	int	Comment identifier
auth	char(15)	Source/originator
subsource	char(8)	Subsystem source
iphase	char(8)	Associated phase
importance	float	Importance of phase
delta	float	Station to event distance
seaz	float	Station to event azimuth
in_wgt	float	Input weight
wgt	float	Output weight
timeres	float	Time residual
azres	float	Azimuth residual
emares	float	Incidence angle residual
slores	float	Slowness residual
vmodelid	int	Velocity model identifier
scorr	float	Station correction in seconds
sdelay	float	Station delay
rflag	char	Human/automatic/finalized observation
ccset	char	Clock correction set
lddate	date	Load date

Amp		Summary information on an amplitude reading
Attribute	***Type***	***Description***
ampid	int	Amplitude identifier
commid	int	Comment identifier
datetime	float	Amplitude date
sta	char(6)	Station code
net	char(8)	Network identifier
auth	char(15)	Source/originator
subsource	char(8)	Subsystem source
channel	char(3)	Channel name
channelsrc	char(8)	Source of channel name
seedchan	char(3)	SEED channel name
location	char(2)	SEED location
iphase	char(8)	Reported phase
amplitude	float	Amplitude
amptype	char(8)	Amplitude type
units	char(4)	Units of amplitude
ampmeas	char	Amplitude measure
eramp	float	Error in amplitude
flagamp	char(4)	Flag to indicate whether amplitude is over P packet,S packet,entire waveform.
per	float	Period
snr	float	Signal to noise ratio
tau	float	Coda duration (F-P time)
quality	float	Subjective quality
rflag	char	Human/automatic/finalized observation
cflag	char(2)	Clip flag
wstart	float	Start of time window
duration	float	Duration of amplitude reading
lddate	date	Load date

AssocAmO		Data associating amplitudes with origins
Attribute	***Type***	***Description***
orid	int	Origin identifier
ampid	int	Amplitude identifier
commid	int	Comment identifier
auth	char(15)	Source/originator
subsource	char(8)	Subsystem source
delta	float	Station to event distance
seaz	float	Station to event azimuth
rflag	char	Human/automatic/finalized observation
lddate	date	Load date

AssocAmM		Data associating amplitudes with magnitudes
Attribute	***Type***	***Description***
<u>magid</u>	int	Magnitude identifier
ampid	int	Amplitude identifier
<u>commid</u>	int	Comment identifier
<u>auth</u>	char(15)	Source/originator
subsource	char(8)	Subsystem source
weight	float	Magnitude weight
in_wgt	float	Input weight
mag	float	Magnitude for this station reading
magres	float	Magnitude residual
magcorr	float	Magnitude correction for this station
importance	float	Importance of amplitude
rflag	char	Human/automatic/finalized observation
<u>lddate</u>	date	Load date

Remark		Comments
Attribute	***Type***	***Description***
<u>commid</u>	int	Comment identifier
<u>lineno</u>	int	Comment line number
<u>remark</u>	char(80)	Free format comment
<u>lddate</u>	date	Load date

Mec		Summary information on a moment tensor
Attribute	***Type***	***Description***
mecid	int	Mechanism identifier
oridin	int	Origin identifier (input)
oridout	int	Origin identifier (output)
<u>magid</u>	int	Mw magnitude identifier
<u>commid</u>	int	Comment identifier
mechtype	char(2)	Mechanism/moment type
mecalgo	char(15)	Mechanism algorithm
scalar	float	Scalar moment
erscalar	float	Error in scalar moment
tft	char(8)	Time function type
tfd	float	Time function duration
mxx	float	m_xx
myy	float	m_yy
mzz	float	m_zz
mxy	float	m_xy
mxz	float	m_xz

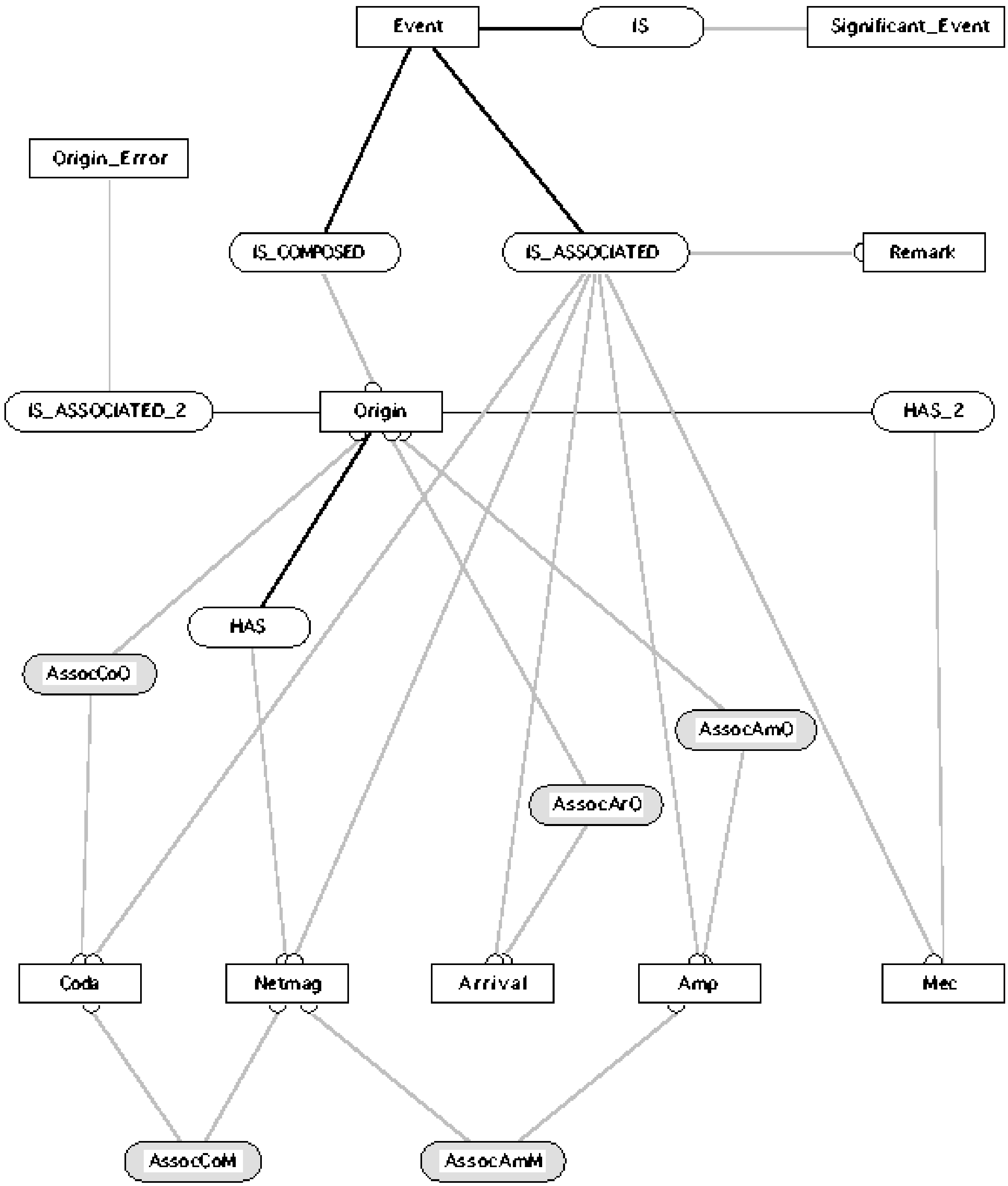
myz	float	m_yz
smxx	float	Uncertainty in m_xx
smyy	float	Uncertainty in m_yy
smzz	float	Uncertainty in m_zz
smxy	float	Uncertainty in m_xy
smxz	float	Uncertainty in m_xz
smyz	float	Uncertainty in m_yz
srcduration	float	Source half-duration
auth	char(15)	Source/origin
subsource	char(8)	Subsystem source
strike1	int	Strike 1 of best double couple
dip1	int	Dip 1 of best double couple
rake1	int	Rake 1 of best double couple
strike2	int	Strike 2 of best double couple
dip2	int	Dip 2 of best double couple
rake2	int	Rake 2 of best double couple
unstrike1	float	Uncertainty in strike 1
undip1	float	Uncertainty in dip 1
unrake1	float	Uncertainty in rake 1
unstrike2	float	Uncertainty in strike 2
undip2	float	Uncertainty in dip 2
unrake2	float	Uncertainty in rake 2
eigenp	float	Eigen pressure
plungep	int	Plunge pressure
strikep	int	Strike pressure
eigenn	float	Eigen null
plungen	int	Plunge null
striken	int	Strike null
eigent	float	Eigen tension
plunget	int	Plunge tension
striket	int	Strike tension
nsta	int	Number of stations used in solution
pvr	int	Percent variance reduction
quality	float	Quality
pdc	int	Percent double couple
pclvd	int	Percent compensated linear vector dipole
piso	int	Percent isotropic
datetime	float	Date solution created
rflag	char	Human/automatic/finalized observation
lddate	date	Load date

Coda		Summary information on a coda
Attribute	***Type***	***Description***
coid	int	Coda identifier
commid	int	Comment identifier
sta	char(6)	Station code
net	char(8)	Network code
auth	char(15)	Source/originator
subsource	char(8)	Subsystem source
channel	char(3)	Channel name
channelsrc	char(8)	Source of channel name
seedchan	char(3)	SEED channel name
location	char(2)	SEED location
codatype	char(3)	Coda type
afix	float	Nominal coda amplitude
afree	float	Free amplitude
qfix	float	Fixed coda decay constant
qfree	float	Free decay
tau	float	Coda duration (F-P time)
nsample	int	Number of code sample windows
rms	float	Residual of mean squares for coda fit
durtype	char(3)	Duration type
iphase	char(8)	Reported phase
eramp	float	Error in amplitude
units	char(4)	Units of amplitude
time1	int	Time for pair 1
amp1	int	Amplitude for pair 1
time2	int	Time for pair 2
amp2	int	Amplitude for pair 2
time3	int	Time for pair 3
amp3	int	Amplitude for pair 3
time4	int	Time for pair 4
amp4	int	Amplitude for pair 4
time5	int	Time for pair 5
amp5	int	Amplitude for pair 5
time6	int	Time for pair 6
amp6	int	Amplitude for pair 6
quality	float	Subjective quality
rflag	char	Human/automatic/finalized observation
lddate	date	Load date

AssocCoM		Data associating codas with magnitudes
Attribute	***Type***	***Description***
magid	int	Magnitude identifier
coid	int	Coda identifier
commid	int	Comment identifier
auth	char(15)	Source/originator
subsource	char(8)	Subsystem source
weight	float	Magnitude weight
in_wgt	float	Input weight
rflag	char	Human/automatic/finalized observation
lddate	date	Load date

AssocCoO		Data associating codas with origins
Attribute	***Type***	***Description***
orid	int	Origin identifier
coid	int	Coda identifier
commid	int	Comment identifier
auth	char(15)	Source/originator
subsource	char(8)	Subsystem source
rflag	char	Human/automatic/finalized observation
lddate	date	Load date

3.- Database relations :



This chapter describes the DBMS relations that comprise the version 1.5 schema. The information given here, along with that in Chapter 4, *Database attributes*, constitutes the data dictionary. There's an entry for each relation. Within the entry, the relation's name appears first, followed by a list of its attributes. A brief description completes the entry.

Keys provide the link by which tables are joined. The following definitions explain the several types of keys.

- A primary key (which can be a concatenation of several attributes) uniquely identifies a row in the table. For example, each **Origin** record is unique by *lat, lon, depth,* and *time*.
- An alternate key also uniquely identifies a row in the table and may be used as the primary key. For example, *orid* may also be used as the primary key for the **Origin** table.
- A foreign key is another table's primary key. Thus, *evid* is a foreign key in the **Origin** table, but is the primary key in the **Event** table. Similarly, *commid* is a foreign key in many of the tables and the primary key in **Remark**.

Name:	**Event**	
Keys:		
	Primary:	<u>*evid*</u>
	Foreign:	<u>*prefor*</u>*, prefmag, prefmec,* <u>*commid*</u>
Data:	<u>*auth*</u>*, subsource,* <u>*etype*</u>*, selectflag,* <u>*lddate*</u>	
Description:	Event to origin connection. The purpose of this relation is to allow the connection of multiple origins to one event. *prefor* points to the preferred origin. *prefmag* points to the preferred magnitude. *prefmec* points to the preferred mechanism.	

Name:	**Significant_Event**	
Keys:		
	Primary:	<u>*evid*</u>
Data:	<u>*evname*</u>*, remarks, nfelt, mmi, pga,* <u>*lddate*</u>	
Description:	This relation stores information about significant events.	

Name:	**Origin**
Keys:	
	Primary: *orid*
	Foreign: *evid*, *prefmag, prefmec*, *commid*
Data:	bogusflag, *datetime*, *lat*, *lon* *depth*, *type*, *algorithm*, *algo_assoc*, *auth*, *subsource, datumhor, datumver, gap, distance, wrms*, *stime*, *erhor, sdep, erlat, erlon, totalarr, totalamp*, *ndef*, *nbs, nbfm, locevid, quality, fdepth, fepi, ftime, vmodelid, cmodelid*, rflag, *lddate*
Description:	Summary of hypocentral parameters. Information describing a derived or reported origin for a particular event is stored in this table.

Name:	**Origin_Error**
Keys:	
	Primary: *orid*
	Foreign:
Data:	*sxx*, *syy*, *szz*, *stt*, *sxy*, *sxz*, *syz*, *stx*, *sty*, *stz*, *azismall, dipsmall, magsmall, aziinter, dipinter, maginter, azilarge, diplarge, maglarge*, *lddate*
Description:	Summary of hypocentral error parameters. This information describes the error data associated with a derived or reported origin for a particular event.

Name:	**Netmag**
Keys:	
	Primary: *magid*
	Foreign: *orid*, *commid*
Data:	*magnitude*, *magtype*, *auth*, *subsource, magalgo, nsta*, *uncertainty*, *gap, distance, quality, rflag*, *lddate*
Description:	This table summarizes estimates of magnitudes of different types for an event. Each magnitude has a unique *magid*.

Name:	**Arrival**
Keys:	
	Primary: <u>*arid*</u>
	Foreign: <u>*commid*</u>
Data:	*datetime*, <u>*sta*</u>, <u>*net*</u>, <u>*auth*</u>, *subsource*, *channel*, *channelsrc*, *seedchan*, *location*, <u>*iphase*</u>, <u>*qual*</u>, *clockqual*, *clockcorr*, *ccset*, <u>*fm*</u>, <u>*ema*</u>, <u>*azimuth*</u>, <u>*slow*</u>, <u>*deltim*</u>, *delinc*, <u>*delaz*</u>, <u>*delslo*</u>, *quality*, <u>*snr*</u>, *rflag*, <u>*lddate*</u>
Description:	Summary information on a seismic arrival. Information characterizing a "seismic phase" observed at a particular station is saved here.

Name:	**AssocArO**
Keys:	
	Primary: <u>*orid*</u>, <u>*arid*</u>
	Foreign: <u>*commid*</u>
Data:	<u>*auth*</u>, *subsource*, <u>*iphase*</u>, *importance*, <u>*delta*</u>, <u>*seaz*</u>, *in_wgt*, <u>*wgt*</u>, <u>*timeres*</u>, <u>*azres*</u>, <u>*emares*</u>, <u>*slores*</u>, *vmodelid*, *scorr*, *sdelay*, *rflag*, *ccset*, <u>*lddate*</u>
Description:	Data associating arrivals with origins. This table has information that connects arrivals (i.e., entries in the **Arrival** relation) to a particular origin. It has a composite key made of *arid* and *orid*.

Name:	**Amp**
Keys:	
	Primary: *ampid*
	Foreign: <u>*commid*</u>
Data:	*datetime*, <u>*sta*</u>, <u>*net*</u>, <u>*auth*</u>, *subsource*, *channel*, *channelsrc*, *seedchan*, *location*, <u>*iphase*</u>, *amplitude*, *amptype*, *units*, *ampmeas*, *eramp*, *flagamp*, <u>*per*</u>, <u>*snr*</u>, *tau*, *quality*, *rflag*, *cflag*, *wstart*, *duration*, <u>*lddate*</u>
Description:	This table contains information about amplitude readings.

Name:	**AssocAmO**
Keys:	Primary: *orid, ampid* Foreign: *commid*
Data:	*auth, subsource, delta, seaz, rflag, lddate*
Description:	Data associating amplitudes with origins. This table has information that connects amplitudes (i.e., entries in the **Amp** relation) to a particular origin. It has a composite key made of *ampid* and *orid*.

Name:	**AssocAmM**
Keys:	Primary: *magid, ampid* Foreign: *commid*
Data:	*auth, subsource, weight, in_wgt, mag, magres, magcorr, importance, rflag, lddate*
Description:	Data associating amplitudes with magnitudes. This table has information that connects amplitudes (i.e., entries in the **Amp** relation) to a particular magnitude. It has a composite key made of *magid* and *ampid*.

Name:	**Remark**
Keys:	Primary: *commid, lineno*
Data:	*remark, lddate*
Description:	Comments. This relation may be used to store free-form comments that embellish records of other relations. The *commid* field in many relations refers to a tuple in the **Remark** table.

Name: **Mec**

Keys:

Primary: *mecid*
Foreign: *magid*, *commid*

Data: *oridin, oridout, mechtype, mecalgo, scalar, erscalar, tft, tfd, mxx, myy, mzz, mxy, mxz, myz, smxx, smyy, smzz, smxy, smxz, smyz, srcduration, auth, subsource, strike1, dip1, rake1, strike2, dip2, rake2, unstrike1, undip1, unrake1, unstrike2, undip2, unrake2, eigenp, plungep, strikep, eigenn, plungen, striken, eigent, plunget, striket, nsta, pvr, quality, pdc, pclvd, piso, datetime, rflag, lddate*

Description: Mechanisms. This relation is used to store information about moment tensors.

Name: **Coda**

Keys:

Primary: *coid*
Foreign: *commid*

Data: *sta, net, auth, subsource, channel, channelsrc, seedchan, location, codatype, afix, afree, qfix, qfree, tau, nsample, rms, durtype, iphase, eramp, units, time1, amp1, time2, amp2, time3, amp3, time4, amp4, time5, amp5, time6, amp6, quality, rflag, lddate*

Description: This relation is used to store information about codas.

Name: **AssocCoM**

Keys:

Primary: *magid*, *coid*
Foreign: *commid*

Data: *auth*, *subsource*, *weight*, *in_wgt*, rflag, *lddate*

Description: Data associating codas with magnitudes. This table has information that connects codas (i.e., entries in the **Coda** relation) to a particular magnitude. It has a composite key made of *magid* and *coid*.

Name:	**AssocCoO**
Keys:	
	Primary: *orid*, *coid*
	Foreign: *commid*
Data:	*auth*, *subsource*, rflag, *lddate*
Description:	Data associating codas with origins. This table has information that connects codas (i.e., entries in the **Coda** relation) to a particular origin. It has a composite key made of *orid* and *coid*.

4.- Database attributes :

This chapter describes each of the attributes used in the version 1.5 schema. Descriptions of the relations are found in Chapter 3, *Database relations*.

Name:	*afix*
Relations:	**Coda**
Description:	Nominal coda amplitude.
Datatype:	***float [number(7,4)]***
NULL Value:	***Yes***
Range:	*afix* > 0.0

Name:	*afree*
Relations:	**Coda**
Description:	Free amplitude.
Datatype:	***float [number(7,4)]***
NULL Value:	***Yes***
Range:	*afree* > 0.0

Name:	*algo_assoc*
Relations:	**Origin**
Description:	Association algorithm used. This is a textual description of the algorithm used for associating the data.
Datatype:	***char(80)***
NULL Value:	***Yes***
Range:	Any free-format string up to 80 characters

Name:	<u>*algorithm*</u>
Relations:	**Origin**
Description:	Location algorithm used. This is a brief textual description of the algorithm used for computing a seismic origin.
Datatype:	***char(15)***
NULL Value:	***Yes***
Range:	Any free-format string up to 15 characters

Name:	*ampid*
Relations:	**Amp**, **AssocAmO**, **AssocAmM**
Description:	Amplitude identification. Each amplitude is assigned a unique positive integer identifying it. This number is used in the **AssocAmO** relation along with the origin identifier to link **Amp** and **Origin**. This number is used also in the **AssocAmM** relation along with the magnitude identifier to link **Amp** and **Netmag**.
Datatype:	***int*** ***[number(15)]***
NULL Value:	***No***
Range:	*ampid* > 0

Name:	*amplitude*
Relations:	**Amp**
Description:	Amplitude in appropriate units for type.
Datatype:	***float***
NULL Value:	***No***
Range:	*amplitude* > 0.0

Name:	*ampmeas*
Relations:	**Amp**
Description:	Amplitude measure.
Datatype:	***char***
NULL Value:	***Yes***
Range:	*ampmeas* ∈ {0 | 1} 0 = peak-to-peak 1 = zero-to-peak

Name:	*amptype*
Relations:	**Amp**
Description:	Amplitude type.
Datatype:	***char(8)***
NULL Value:	***Yes***
Range:	*amptype* ∈ {WA | WAS | PGA | PGV | PGD | WAC | WAU | IV2 | SP.3 | SP1.0 | SP3.0 | ML100 | ME100 | EGY} WA = Wood-Anderson photographic WAS = Wood-Anderson synthetic PGA = peak ground acceleration PGV = peak ground velocity PGD = peak ground displacement WAC = Wood-Anderson corrected WAU = Wood-Anderson uncorrected IV2 = integral of velocity squared SP.3 = spectral peak SP1.0 = spectral peak SP3.0 = spectral peak ML100 = local magnitude ME100 = energy magnitude EGY = energy

Name:	*amp1, amp2, amp3, amp4, amp5, amp6*
Relations:	***Coda***
Description:	Amplitude for pairs.
Datatype:	***int*** ***[number(10)]***
NULL Value:	***Yes***
Range:	*amp1, amp2, amp3, amp4, amp5, amp6* > 0

Name:	<u>*arid*</u>
Relations:	**Arrival**, **AssocArO**
Description:	Arrival identifier. Each arrival is assigned a unique positive integer identifying it. This number is used in the **AssocArO** relation along with the origin identifier to link **Arrival** and **Origin**.
Datatype:	***int*** ***[number(15)]***
NULL Value:	***No***
Range:	*arid* > 0

Name:	<u>*auth*</u>
Relations:	**Event**, **Origin**, **Netmag**, **Arrival**, **Amp**, **Mec**, **Coda, AssocArO, AssocAmO, AssocAmM, AssocCoM, AssocCoO**
Description:	The *auth* field specifies the source of the information. This may be a network identifier (FDSN or otherwise), or any other string indicating the source of the information in the record.
Datatype:	***char(15)***
NULL Value:	***No***
Range:	Any free-format string up to 15 characters

Name:	*aziinter*
Relations:	**Origin_Error**
Description:	Azimuth of intermediate principal error.
Datatype:	***float*** ***[number(5,2)]***
NULL Value:	***Yes***
Range:	0.0 ≤ *aziinter* ≤ 360.0
Units:	***Degrees***

Name:	*azilarge*
Relations:	**Origin_Error**
Description:	Azimuth of largest principal error.
Datatype:	***float*** ***[number(5,2)]***
NULL Value:	***Yes***
Range:	0.0 ≤ *azilarge* ≤ 360.0
Units:	***Degrees***

Name:	*azimuth*
Relations:	**Arrival**
Description:	Observed azimuth. This is the estimated station-to-event azimuth measured clockwise from north.
Datatype:	***float*** ***[number(4,1)]***
NULL Value:	***Yes***
Range:	0.0 ≤ *azimuth* ≤ 360.0
Units:	***Degrees***

Name:	*azismall*
Relations:	**Origin_Error**
Description:	Azimuth of smallest principal error.
Datatype:	***float*** ***[number(5,2)]***
NULL Value:	***Yes***
Range:	0.0 ≤ *azismall* ≤ 360.0
Units:	***Degrees***

Name:	*azres*
Relations:	**AssocArO**
Description:	Azimuth residual. This is the difference between the measured station-to-event azimuth for an arrival and the true azimuth. The "true" azimuth is the bearing to the inferred event origin.
Datatype:	***float*** ***[number(5,3)]***
NULL Value:	***Yes***
Range:	-180.0 ≤ *azres* ≤ 180.0
Units:	***Degrees***

Name:	*bogusflag*
Relations:	**Origin**
Description:	This field describes whether an origin is bogus or not.
Datatype:	***int*** ***[number(1)]***
NULL Value:	***No***
Range:	*bogusflag* ∈ {0 | 1} 0 = no 1 = yes

Name:	*ccset*
Relations:	**Arrival, AssocArO**
Description:	This field describes whether the clock correction has been added or applied to the arrivals in computing a hypocenter solution.
Datatype:	***int*** ***[number(1)]***
NULL Value:	***Yes***
Range:	*ccset* ∈ {0 | 1} 0 = no 1 = yes

Name:	*cflag*
Relations:	**Amp**
Description:	This flag indicates whether the amplitude value is below noise, on scale or clipped.
Datatype:	***char(2)***
NULL Value:	***Yes***
Range:	*cflag* ∈ { BN | OS | CL} BN = below noise OS = on scale CL = clipped

Name:	*channel*
Relations:	**Arrival**, **Amp**, **Coda**
Description:	Channel name.
Datatype:	***char(3)***
NULL Value:	***Yes***
Range:	Any valid channel name, which may be network-specific. Networks using SEED channel names should use the same value for *channel* and *seedchan*

Name:	*channelsrc*
Relations:	**Arrival**, **Amp**, **Coda**
Description:	Domain for channel. This specifies what naming convention is used for the channel name (i.e. SEED, USGS, etc.).
Datatype:	***char(8)***
NULL Value:	***Yes***
Range:	Any free-format string up to 8 characters long

Name:	*clockcorr*
Relations:	**Arrival**
Description:	This field indicates the amount of time to add to the arrival time if clock correction is known.
Datatype:	***int***
NULL Value:	***Yes***
Units:	***Microseconds***

Name:	*clockqual*
Relations:	**Arrival**
Description:	This field describes whether the clock quality associated with a particular arrival is known, and if so, is good or bad.
Datatype:	***char***
NULL Value:	***Yes***
Range:	*clockqual* ∈ {U | G | B} U = unknown G = good B = bad

Name:	*cmodelid*
Relations:	**Origin**
Description:	Crustal model identifier.
Datatype:	***int*** ***[number(3)]***
NULL Value:	***Yes***
Range:	*cmodelid* > 0

Name:	*codatype*
Relations:	**Coda**
Description:	Coda type.
Datatype:	***char(3)***
NULL Value:	***Yes***
Range:	*codatype* ∈ {P | S} P = USGS P-wave coda S = UCB S-wave coda

Name:	*coid*
Relations:	**Coda**, **AssocCoM**, **AssocCoO**
Description:	Coda identification. Each coda is assigned a unique positive integer which identifies it in the database.
Datatype:	***int*** ***[number(15)]***
NULL Value:	***No***
Range:	*coid* > 0

Name:	<u>*commid*</u>
Relations:	**Event**, **Origin**, **Netmag**, **Arrival**, **AssocArO**, **Amp**, **AssocAmO**, **AssocAmM**, **Remark**, **Mec**, **Coda**, **AssocCoM**, **AssocCoO**
Description:	Comment identification. This is a key used to point to free-form comments entered in the **Remark** relation. These comments store additional information about a tuple in another relation. Within the **Remark** relation, there may be many tuples with the same *commid* and different *lineno*, but the same *commid* will appear in only one other tuple among the rest of the relations in the database.
Datatype:	***int*** ***[number(15)]***
NULL Value:	***Yes*** (***No*** in the **Remark** relation)
Range:	*commid* > 0

Name:	*datetime*
Relations:	**Origin**, **Arrival**, **Amp**, **Mec**
Description:	The date of associated with information in the record, in true epoch format.
Datatype:	***float*** ***[number(25,10)]***
NULL Value:	**No**
Range:	Any valid float. *datetime* = 0.0 on **January 01, 1970** at **00:00:00.0000**
Units:	***Seconds***

Name:	*datumhor*
Relations:	**Origin**
Description:	Datum type for horizontal (latitude and longitude) coordinates.
Datatype:	***char(8)***
NULL Value:	***Yes***
Range:	*datumhor* ∈ { NAD27 | WGS84}

Name:	*datumver*
Relations:	**Origin**
Description:	Datum type for depth or elevation.
Datatype:	***char(8)***
NULL Value:	***Yes***
Range:	*datumver* ∈ { NAD27 | WGS84 | AVERAGE} AVERAGE = average of stations

Name:	*delaz*
Relations:	**Arrival**
Description:	Uncertainty in the azimuth. This attribute gives the standard deviation of the azimuth of a signal.
Datatype:	***float [number(5,2)]***
NULL Value:	***Yes***
Range:	*delaz* > 0.0
Units:	***Degrees***

Name:	*delinc*
Relations:	**Arrival**
Description:	Uncertainty in the angle. This attribute gives the standard deviation of the angle of emergence.
Datatype:	***float [number(4,2)]***
NULL Value:	***Yes***
Range:	*delinc* ≥ 0.0
Units:	***Degrees***

Name:	<u>*delslo*</u>
Relations:	**Arrival**
Description:	Uncertainty in slowness. This attribute gives the standard deviation of the slowness of a signal.
Datatype:	***float [number(8,4)]***
NULL Value:	***Yes***
Range:	*delslo* > 0.0
Units:	***Seconds/Kilometers***

Name:	<u>*delta*</u>
Relations:	**AssocArO**, **AssocAmO**
Description:	Source-receiver distance. This attribute is the arc length, over the earth's surface, of the path the seismic phase follows from source to receiver. The location of the origin is specified in the origin record referenced by the attribute *orid*. The attribute *arid* points to the record in the **Arrival** relation that identifies the receiver. The value of the attribute can exceed 180 degrees, it can even exceed 360 degrees, the geographic distance between source and receiver is *delta* mod(180).
Datatype:	***float [number(5,1)]***
NULL Value:	***Yes***
Range:	*delta* ≥ 0.0
Units:	***Degrees***

Name:	*deltim*
Relations:	**Arrival**
Description:	Arrival time uncertainty. This attribute gives the standard deviation of a detection time.
Datatype:	***float [number(5,2)]***
NULL Value:	***Yes***
Range:	*deltim* ≥ 0.0
Units:	***Seconds***

Name:	*depth*
Relations:	**Origin**
Description:	Source depth. This attribute gives the depth of the event origin. The reference depth of 0 is usually defined as the average elevation of stations used to determine the origin, although other data may be specified. Positive values are down, negative values are up from the reference point.
Datatype:	***float [number(7,3)]***
NULL Value:	***Yes***
Range:	-10.0 ≤ *depth* ≤ 1000.0
Units:	***Kilometers***

Name:	*dipinter*
Relations:	**Origin_Error**
Description:	Dip of intermediate principal error.
Datatype:	***float [number(4,2)]***
NULL Value:	***Yes***
Range:	-90.0 ≤ *dipinter* ≤ 90.0
Units:	***Degrees***

Name:	*diplarge*
Relations:	**Origin_Error**
Description:	Dip of largest principal error.
Datatype:	***float* [number(4,2)]**
NULL Value:	***Yes***
Range:	-90.0 ≤ *diplarge* ≤ 90.0
Units:	***Degrees***

Name:	*dipsmall*
Relations:	**Origin_Error**
Description:	Dip of smallest principal error.
Datatype:	***float* [number(4,2)]**
NULL Value:	***Yes***
Range:	-90.0 ≤ *dipsmall* ≤ 90.0
Units:	***Degrees***

Name:	*dip1*
Relations:	**Mec**
Description:	Dip of fault plane 1 from the best double couple.
Datatype:	***int* [number(2)]**
NULL Value:	***Yes***
Range:	-90 ≤ *dip1* ≤ 90
Units:	***Degrees***

Name:	*dip2*
Relations:	**Mec**
Description:	Dip of fault plane 2 from the best double couple.
Datatype:	***int*** ***[number(2)]***
NULL Value:	***Yes***
Range:	-90 ≤ *dip2* ≤ 90
Units:	***Degrees***

Name:	*distance*
Relations:	**Origin**, **Netmag**
Description:	Distance from the origin to the nearest station.
Datatype:	***float*** ***[number(7,3)]***
NULL Value:	***Yes***
Range:	*distance* ≥ 0.0
Units:	***Kilometers***

Name:	*duration*
Relations:	**Amp**
Description:	Duration of amplitude reading.
Datatype:	***float***
NULL Value:	***No***
Range:	*duration* > 0.0
Units:	***Seconds***

Name:	*durtype*
Relations:	**Coda**
Description:	Duration type.
Datatype:	***char(3)***
NULL Value:	***Yes***
Range:	*durtype* ∈ {S} S = S-wave

Name:	*eigenn*
Relations:	**Mec**
Description:	Eigen value of null axis in the principal axes representation of the best double couple.
Datatype:	***float [number(7,5)]***
NULL Value:	***Yes***

Name:	*eigenp*
Relations:	**Mec**
Description:	Eigen value of compressional axis in the principal axes representation of the best double couple.
Datatype:	***float [number(7,5)]***
NULL Value:	***Yes***

Name:	*eigent*
Relations:	**Mec**
Description:	Eigen value of tension axis in the principal axes representation of the best double couple.
Datatype:	***float [number(7,5)]***
NULL Value:	***Yes***

Name:	<u>*ema*</u>
Relations:	**Arrival**
Description:	Emergence angle. This attribute is the emergence angle of an arrival, as observed at a three-component station or array. The value increases from the vertical direction towards the horizontal.
Datatype:	***float*** ***[number(5,2)]***
NULL Value:	***Yes***
Range:	$0.0 \leq ema \leq 90.0$
Units:	***Degrees***

Name:	<u>*emares*</u>
Relations:	**AssocArO**
Description:	Emergence angle residual. This attribute is the difference between an observed emergence angle and the theorical prediction for the same phase, assuming an event location as specified by the accompanying *orid*.
Datatype:	***float*** ***[number(5,3)]***
NULL Value:	***Yes***
Range:	$-90.0 \leq emares \leq 90.0$
Units:	***Degrees***

Name:	*eramp*
Relations:	**Amp**, **Coda**
Description:	Uncertainty in amplitude measurement.
Datatype:	***float*** ***[number(5,3)]***
NULL Value:	***Yes***
Range:	$eramp \geq 0.0$

Name:	*erhor*
Relations:	**Origin**
Description:	Horizontal error. This attribute denotes the location uncertainty that accompanies the location.
Datatype:	***float*** ***[number(7,3)]***
NULL Value:	***Yes***
Range:	*erhor* ≥ 0.0
Units:	***Kilometers***

Name:	*erlat*
Relations:	**Origin**
Description:	Error in latitude. This is the standard deviation of the latitude estimate.
Datatype:	***float*** ***[number(7,3)]***
NULL Value:	***Yes***
Range:	*erlat* ≥ 0.0
Units:	***Kilometers***

Name:	*erlon*
Relations:	**Origin**
Description:	Error in longitude. This is the standard deviation of the longitude estimate.
Datatype:	***float*** ***[number(7,3)]***
NULL Value:	***Yes***
Range:	*erlon* ≥ 0.0
Units:	***Kilometers***

Name:	*erscalar*
Relations:	***Mec***
Description:	Error in scalar moment. This is the standard deviation of the scalar moment.
Datatype:	***float* [*number(3,2)*]**
NULL Value:	***Yes***
Range:	*erscalar* ≥ 0.0
Units:	***Dyne centimeters***

Name:	*etype*
Relations:	**Event**
Description:	This attribute is used to identify the type of seismic event, when known.
Datatype:	***char(7)***
NULL Value:	***Yes***
Range:	*etype* ∈ {le | re | ts | qb | nt | uk | sn} le = local event re = regional event ts = teleseism qb = quarry blast nt = nuclear test uk = unknown event sn = sonic blast

Name:	*evid*
Relations:	**Event**, **Significant_Event**, **Origin**
Description:	Event identifier. Each event is assigned a unique integer which identifies it in a database. It is possible for several records in the **Origin** relation to have the same *evid*. This indicates there are several opinions about the location of the event. The *evid* does not have to be identical to the local network's event identifier, which is stored in the **Origin** table. This allows merging data for a single event from multiple sources without insisting that database *evid* is identical to the source's local event identifier.
Datatype:	***int*** ***[number(15)]***
NULL Value:	***No***
Range:	*evid* > 0

Name:	*evname*
Relations:	**Significant_Event**
Description:	Event name. This is a descriptive name for major events.
Datatype:	***char(80)***
NULL Value:	***Yes***
Range:	Any free-format string up to 80 characters

Name:	*fdepth*
Relations:	**Origin**
Description:	This attribute indicates if the depth was fixed or not to compute the solution.
Datatype:	***char***
NULL Value:	***Yes***
Range:	*fdepth* ∈ {y | n} y = fixed depth n = free depth

Name:	*fepi*
Relations:	**Origin**
Description:	This attribute indicates if the epicenter was fixed or not to compute the solution.
Datatype:	***char***
NULL Value:	***Yes***
Range:	*fepi* ∈ {y | n} y = fixed epicenter n = free epicenter

Name:	*flagamp*
Relations:	**Amp**
Description:	This attribute is a flag to indicate whether amplitude is over P packet, S packet, entire waveform, etc.
Datatype:	***char(4)***
NULL Value:	***Yes***
Range:	*flagamp* ∈ {SUR | P | S | ALL} SUR = surface wave P = P packet S = S packet ALL = entire waveform

Name:	*fm*
Relations:	**Arrival**
Description:	First motion. This is a two-character indication of first motion. The first character describes first motion seen on short-period channels and the second holds for long-period instruments. Compression (dilation) on a short-period sensor is denoted by c(d) and compression (dilation) on a long-period sensor is denoted by u(r). Empty character positions will be indicated by dots (e.g., ".r").
Datatype:	***char(2)***
NULL Value:	***Yes***
Range:	All two-letter permutations of (c | d | .} ∪ {u | r | .}

Name:	*ftime*
Relations:	**Origin**
Description:	This attribute indicates if the origin time was fixed or not to compute the solution.
Datatype:	***char***
NULL Value:	***Yes***
Range:	*ftime* ∈ {y | n} y = fixed origin time n = free origin time

Name:	*gap*
Relations:	**Origin**, **Netmag**
Description:	Gap in azimuthal coverage.
Datatype:	***float [number(4,1)]***
NULL Value:	***Yes***
Range:	$0.0 \le gap \le 360.0$
Units:	***Degrees***

Name:	*in_wgt*
Relations:	**AssocArO**, **AssocAmM**, **AssocCoM**
Description:	Input weight.
Datatype:	***float*** ***[number(4,3)]***
NULL Value:	***Yes***
Range:	$0.0 < in\_wgt \leq 1.0$ 0.0 = no weight (not used)

Name:	*importance*
Relations:	**AssocArO**, **AssocAmM**
Description:	This attribute denotes the importance of a phase/amplitude reading towards an origin/magnitude. 0.0 means no importance, 1.0 implies extremely important.
Datatype:	***float*** ***[number(2,1)]***
NULL Value:	***Yes***
Range:	$0.0 < importance \leq 1.0$

Name:	*iphase*
Relations:	**Arrival**, **AssocArO**, **Coda**
Description:	Reported phase. This eight-character field holds the name initially given to a seismic phase. Standard seismological labels for the types of signals (or phases) are used (e.g., P, PKP, PcP, pP). Both upper and lower case letters are available and should be used when appropriate, for example, pP or PcP.
Datatype:	***char(8)***
NULL Value:	***Yes***
Range:	Any free-format string up to 8 characters

Name:	*lat*
Relations:	**Origin**
Description:	Latitude. This attribute is the geographic latitude. Locations north of the equator have positive latitudes.
Datatype:	***float*** ***[number(9,7)]***
NULL Value:	***No***
Range:	$-90.0 \leq lat \leq 90.0$
Units:	***Degrees***

Name:	*lddate*
Relations:	**Event**, **Significant_Event**, **Origin**, **Origin_Error**, **Netmag**, **Arrival**, **AssocArO**, **Amp**, **AssocAmO**, **AssocAmM**, **Remark**, **Mec**, **Coda**, **AssocCoM**, **AssocCoO**
Description:	Load date. Date and time that the record was created or last modified, in Oracle ***date*** datatype.
Datatype:	***date***
NULL Value:	***No***
Range:	Any valid date between **January 01, 4712 BC** and **January 01, 4712 AD**
Units:	***YYYY/MM/DD HH24:MI:SS***

Name:	*lineno*
Relations:	**Remark**
Description:	Comment line number. This integer attribute is assigned as a sequence number for multiple line comments. The combination of *commid* and *lineno* is unique.
Datatype:	***int***
NULL Value:	***No***
Range:	$lineno > 0$

Name:	*location*
Relations:	**Arrival**, **Amp**, **Coda**
Description:	Describes the individual sites on an array station, operated by the same network operator.
Datatype:	***char(2)***
NULL Value:	***Yes***
Range:	Any free-format string up to 2 characters long

Name:	*locevid*
Relations:	**Origin**
Description:	Local event identifier. This attribute describes the original event identifier of the event. In combination with *auth* and *subsource*, it can be used to uniquely identify the original event identifier from that source. For example, waveforms from the original source may be associated with this identifier.
Datatype:	***char(12)***
NULL Value:	***Yes***
Range:	Any free-format string up to 12 characters long

Name:	<u>lon</u>
Relations:	**Origin**
Description:	Longitude. This attribute is the geographic longitude. Longitudes are measured positive east of the Greenwich meridian.
Datatype:	***float [number(10,7)]***
NULL Value:	***No***
Range:	$-180.0 \leq lon \leq 180.0$
Units:	***Degrees***

Name:	*mag*
Relations:	**AssocAmM**
Description:	Magnitude for this station reading.
Datatype:	***float [number(5,2)]***
NULL Value:	***Yes***
Range:	-10.0 < *mag* < 10.0
Units:	***Magnitude units***

Name:	*magalgo*
Relations:	**Netmag**
Description:	This attribute represents the algorithm used to compute the magnitude.
Datatype:	***char(15)***
NULL Value:	***Yes***
Range:	Any free-format string up to 15 characters

Name:	*magcorr*
Relations:	**AssocAmM**
Description:	Magnitude correction for this channel.
Datatype:	***float [number(5,2)]***
NULL Value:	***Yes***
Range:	-10.0 ≤ *magcorr* ≤ 10.0
Units:	***Magnitude units***

Name:	<u>*magid*</u>
Relations:	**Netmag**, **AssocAmM**, **Mec**, **AssocCoM**
Description:	Magnitude identifier. This key is used to identify a magnitude in the **Netmag** relation. It is required for every magnitude.
Datatype:	***int*** ***[number(15)]***
NULL Value:	***No***
Range:	*magid* > 0

Name:	*maginter*
Relations:	**Origin_Error**
Description:	Magnitude of intermediate principal error.
Datatype:	***float*** ***[number(7,5)]***
NULL Value:	***Yes***
Range:	*maginter* ≥ 0.0
Units:	***Kilometers***

Name:	*maglarge*
Relations:	**Origin_Error**
Description:	Magnitude of largest principal error.
Datatype:	***float*** ***[number(7,5)]***
NULL Value:	***Yes***
Range:	*maglarge* ≥ 0.0
Units:	***Kilometers***

Name:	*magnitude*
Relations:	**Netmag**
Description:	Magnitude. This gives the magnitude value of the type indicated by *magtype*.
Datatype:	***float [number(5,2)]***
NULL Value:	***No***
Range:	-10.0 < *magnitude* < 10.0
Units:	***Magnitude units***

Name:	*magres*
Relations:	**AssocAmM**
Description:	Magnitude residual.
Datatype:	***float [number(5,2)]***
NULL Value:	***Yes***
Range:	*magres* ≥ 0.0
Units:	***Magnitude units***

Name:	*magsmall*
Relations:	**Origin_Error**
Description:	Magnitude of smallest principal error.
Datatype:	***float [number(7,5)]***
NULL Value:	***Yes***
Range:	*magsmall* ≥ 0.0
Units:	***Kilometers***

Name: *magtype*

Relations: **Netmag**

Description: Magnitude type. This character string is used to specify the type of the magnitude measure.

Datatype: ***char(6)***

NULL Value: ***No***

Range: *magtype* ∈ {a | b | e | l | l1 | l2 | lg | c | s | w | z | B | un | d | h | n}
a = primary amplitude magnitude (Jerry Eaton's XMAG)
b = body-wave magnitude
e = energy magnitude
l = local magnitude
l1 = traditional UCB local magnitude
l2 = network UCB local magnitude
lg = Lg magnitude
c = primary coda magnitude
s = Ms – surface wave magnitude
w = moment magnitude
z = low gain (Z component) magnitude of Hirshorn and Lindh (1989)
B = magnitude estimated from 14-kg Benioff's
un = unknown magnitude type
d = duration magnitude
h = helicorder magnitude (CIT, short-period Benioff)
n = no magnitude

Name: *mecid*

Relations: **Mec**

Description: Mechanism identification. Each mechanism is assigned a unique positive integer which identifies it in the database.

Datatype: ***int*** ***[number(15)]***

NULL Value: ***No***

Range: *mecid* > 0

Name:	*mecalgo*
Relations:	**Mec**
Description:	This attribute represents the algorithm used to compute the mechanism.
Datatype:	***char(15)***
NULL Value:	***Yes***
Range:	Any free-format string up to 15 characters

Name:	*mechtype*
Relations:	**Mec**
Description:	Mechanism/moment type.
Datatype:	***char(2)***
NULL Value:	***Yes***
Range:	*mechtype* ∈ {C | F | H | N | S} C = complete-waveform inversion F = first motion mechanism H = half duration of S-wave N = near-field inversion S = surface-wave inversion

Name:	*mmi*
Relations:	**Significant_Event**
Description:	Reported or inferred Mercali intensity for event.
Datatype:	***int*** ***[number(2)]***
NULL Value:	***Yes***
Range:	$1 \leq mmi \leq 12$

Name:	*mxx, myy, mzz, mxy, mxz, myz*
Relations:	**Mec**
Description:	Moment tensor elements.
Datatype:	***float [number(7,5)]***
NULL Value:	***Yes***
Range:	$-10^{32} < mxx, myy, mzz, mxy, mxz, myz < 10^{32}$
Units:	***Dyne centimeters***

Name:	*nbfm*
Relations:	**Origin**
Description:	Number of P first motions with non-zero input weights.
Datatype:	***int [number(4)]***
NULL Value:	***Yes***
Range:	$nbfm \geq 0$

Name:	*nbs*
Relations:	**Origin**
Description:	Number of S phase readings with non-zero output weights.
Datatype:	***int [number(4)]***
NULL Value:	***Yes***
Range:	$nbs \geq 0$

Name:	<u>*ndef*</u>
Relations:	**Origin**
Description:	This attribute denotes the number of phases with non-zero output weights used in hypocenter calculation.
Datatype:	***int*** ***[number(5)]***
NULL Value:	***Yes***
Range:	*ndef* ≥ 0

Name:	<u>*net*</u>
Relations:	**Arrival**, **Amp**, **Coda**
Description:	Unique network identifier. This character string is the name of a seismic network.
Datatype:	***char(8)***
NULL Value:	***Yes***
Range:	Any free-format string up to 8 characters. If available, use the two character FDSN code for the network

Name:	*nfelt*
Relations:	**Significant_Event**
Description:	Number of felt reports.
Datatype:	***int***
NULL Value:	***Yes***
Range:	*nfelt* > 0

Name:	*nsample*
Relations:	**Coda**
Description:	Number of coda sample windows.
Datatype:	***int*** ***[number(6)]***
NULL Value:	***Yes***
Range:	*nsample* > 0

Name:	*nsta*
Relations:	**Netmag**, **Mec**
Description:	This quantity is the number of observations with non-zero input weights used to compute the magnitude or mechanism..
Datatype:	***int*** ***[number(5)]***
NULL Value:	***Yes***
Range:	*nsta* > 0

Name:	*orid*
Relations:	**Origin**, **Origin_Error**, **Netmag**, **AssocArO**, **AssocAmO**, **AssocCoO**
Description:	Origin identification. Each origin is assigned a unique positive integer which identifies it in the database. The *orid* is used to identify one of the many hypotheses of the actual location of the event.
Datatype:	***int*** ***[number(15)]***
NULL Value:	***No***
Range:	*orid* > 0

Name:	*oridin*
Relations:	**Mec**
Description:	Input origin. This attribute holds the origin identifier, *orid*, that points to the origin from which the mechanism was computed.
Datatype:	***int*** ***[number(15)]***
NULL Value:	***Yes***
Range:	*oridin > 0*

Name:	*oridout*
Relations:	**Mec**
Description:	Output origin. This attribute holds the origin identifier, *orid*, that points to the origin computed from the mechanism.
Datatype:	***int*** ***[number(15)]***
NULL Value:	***Yes***
Range:	*oridout > 0*

Name:	*pclvd*
Relations:	**Mec**
Description:	Percent compensated linear vector dipole of the seismic moment tensor.
Datatype:	***int*** ***[number(5,2)]***
NULL Value:	***Yes***
Range:	$0 \leq pclvd \leq 100$

Name:	*pdc*
Relations:	**Mec**
Description:	Percent double couple of the seismic moment tensor.
Datatype:	***int*** ***[number(5,2)]***
NULL Value:	***Yes***
Range:	$0 \leq pdc \leq 100$

Name:	*per*
Relations:	**Amp**
Description:	Signal period. This attribute is the period of the signal described by the *amplitude* record.
Datatype:	***float*** ***[number(10,4)]***
NULL Value:	***Yes***
Range:	$per > 0.0$
Units:	***Seconds***

Name:	*pga*
Relations:	**Significant_Event**
Description:	Peak ground acceleration recorded by any station for the event. Represented in g, where 1.0 = 1g.
Datatype:	***float*** ***[number(6,4)]***
NULL Value:	***Yes***
Range:	$pga > 0.0$
Units:	***g***

Name:	*piso*
Relations:	**Mec**
Description:	Percent isotropic of the seismic moment tensor.
Datatype:	***int*** ***[number(5,2)]***
NULL Value:	***Yes***
Range:	$0 \leq piso \leq 100$

Name:	*plungen*
Relations:	**Mec**
Description:	Plunge of the null axis in the principal axes representation of the best double couple.
Datatype:	***int*** ***[number(2)]***
NULL Value:	***Yes***
Range:	$0 \leq plungen \leq 90$
Units:	***Degrees***

Name:	*plungep*
Relations:	**Mec**
Description:	Plunge of the compressional axis in the principal axes representation of the best double couple.
Datatype:	***int*** ***[number(2)]***
NULL Value:	***Yes***
Range:	$0 \leq plungep \leq 90$
Units:	***Degrees***

Name:	*plunget*
Relations:	**Mec**
Description:	Plunge of the tension axis in the principal axes representation of the best double couple.
Datatype:	***int*** ***[number(2)]***
NULL Value:	***Yes***
Range:	$0 \leq plunget \leq 90$
Units:	***Degrees***

Name:	*prefmag*
Relations:	**Event**, **Origin**
Description:	Preferred magnitude. This attribute holds the magnitude identifier, *magid*, that points to the preferred magnitude for a seismic event or an origin.
Datatype:	***int*** ***[number(15)]***
NULL Value:	***Yes***
Range:	$prefmag > 0$

Name:	*prefmec*
Relations:	**Event**, **Origin**
Description:	Preferred mechanism. This attribute holds the mechanism identifier, *mecid*, that points to the preferred mechanism for a seismic event or an origin.
Datatype:	***int*** ***[number(15)]***
NULL Value:	***Yes***
Range:	$prefmec > 0$

Name:	*prefor*
Relations:	**Event**
Description:	Preferred origin. This attribute holds the origin identifier, *orid*, that points to the preferred origin for a seismic event.
Datatype:	***int*** ***[number(15)]***
NULL Value:	***Yes*** (only if no origin exists for the event)
Range:	*prefor* > 0

Name:	*pvr*
Relations:	**Mec**
Description:	Percent variance reduction.
Datatype:	***float*** ***[number(5,2)]***
NULL Value:	***Yes***
Range:	0.0 ≤ *pvr* ≤ 100.0

Name:	*qfix*
Relations:	**Coda**
Description:	Fixed coda decay constant.
Datatype:	***float*** ***[number(7,4)]***
NULL Value:	***Yes***
Range:	Any valid float

Name:	*qfree*
Relations:	**Coda**
Description:	Free decay.
Datatype:	***float [number(7,4)]***
NULL Value:	***Yes***
Range:	Any valid float

Name:	*qual*
Relations:	**Arrival**
Description:	Onset quality. This single-character flag is used to denote the sharpness of the onset of a seismic phase. This attribute does not denote the timing accuracy.
Datatype:	***char***
Null Value:	***Yes***
Range:	*qual* $\in$ {i | e | w} i = impulsive e = emergent w = weak

Name:	*quality*
Relations:	**Origin**, **Netmag**, **Arrival**, **Amp**, **Mec**, **Coda**
Description:	This attribute denotes the quality of an origin, an arrival, or a mechanism. 1.0 = good, 0.0 = bad.
Datatype:	***float [number(2,1)]***
NULL Value:	***Yes***
Range:	$0.0 \leq quality \leq 1.0$

Name:	*rake1*
Relations:	**Mec**
Description:	Rake of fault plane 1 from the best double couple solution.
Datatype:	***int*** ***[number(4)]***
NULL Value:	***Yes***
Range:	-180 ≤ *rake1* ≤ 180
Units:	***Degrees***

Name:	*rake2*
Relations:	**Mec**
Description:	Rake of fault plane 2 from the best double couple solution.
Datatype:	***int*** ***[number(4)]***
NULL Value:	***Yes***
Range:	-180 ≤ *rake2* ≤ 180
Units:	***Degrees***

Name:	*remark*
Relations:	**Remark**
Description:	Descriptive text. This single line of text is an arbitrary comment about a record in the database. The comment is linked to its "parent" relation only by forward reference from *commid* in the tuple of the relation of interest.
Datatype:	***char(80)***
NULL Value:	***Yes***
Range:	Any free-format string up to 80 characters long

Name:	*remarks*
Relations:	**Significant_Event**
Description:	This attribute is used to give some information about the event, such as felt or damage.
Datatype:	***char(2)***
NULL Value:	***Yes***
Range:	*remarks* ∈ {f | d} f = felt d = damage

Name:	*rflag*
Relations:	**Origin, Arrival, Amp, Coda, Netmag, Mec, AssocCoO, AssocCoM, AssocAmM, AssocAmO, AssocArO**
Description:	This flag describes whether an observation was generated automatically, by a human, or has been finalized.
Datatype:	***char***
NULL Value:	***No***
Range:	*rflag* ∈ {A | H | F} A = automatic H = reviewed by human F = finalized

Name:	*rms*
Relations:	**Coda**
Description:	Square root of the sum of the squares of the coda residuals, divided by the number of observations.
Datatype:	***float [number(5,3)]***
NULL Value:	***Yes***
Range:	*rms* ≥ 0.0

Name:	*scalar*
Relations:	***Mec***
Description:	Seismic scalar moment.
Datatype:	***float [number(6,4)]***
NULL Value:	***Yes***
Range:	$-10^{32} < scalar < 10^{32}$
Units:	***Dyne centimeters***

Name:	*scorr*
Relations:	**AssocArO**
Description:	Station correction.
Datatype:	***float [number(6,4)]***
NULL Value:	***Yes***
Range:	Any valid float
Units:	***Seconds***

Name:	*sdelay*
Relations:	**AssocArO**
Description:	Station delay.
Datatype:	***float [number(7,4)]***
NULL Value:	***Yes***
Range:	Any valid float
Units:	***Seconds***

Name:	*sdep*
Relations:	**Origin**
Description:	Depth error. This is the standard deviation of the depth estimate.
Datatype:	***float [number(6,3)]***
NULL Value:	***Yes***
Range:	*sdep* ≥ 0.0
Units:	***Kilometers***

Name:	*seaz*
Relations:	**AssocArO**, **AssocAmO**
Description:	Station to event azimuth. This attribute is calculated from the station and event locations. It is measured clockwise from North.
Datatype:	***float [number(4,1)]***
NULL Value:	***Yes***
Range:	0.0 ≤ *seaz* ≤ 360.0
Units:	***Degrees***

Name: *seedchan*

Relations: **Arrival**, **Amp**, **Coda**

Description: SEED channel name. The first character denotes the band code, the second for the instrument code and the third represents the component code.

Datatype: ***char(3)***

NULL Value: ***Yes***

Range: *seedchan* $\in$ {E | S | H | B | M | L | V | U | R} $\cup$
{A | B | D | F | G | H | I | K | L | M | P | R | S | V | T | W} $\cup$
{Z | N | E | A | B | C | T | R | 1 | 2 | 3 | U | V | W}

Band code:

	Sample rate (Hz)	Corner period (sec)
E = extremely short period	≥ 80	< 10
S = short period	≥ 10 to < 80	< 10
H = high broad band	≥ 80	≥ 10
B = broad band	≥ 10 to < 80	≥ 10
M = mid period	> 1 to < 10	
L = long period	~ 1	
V = very long period	~ 0.1	
U = ultra long period	~ 0.01	
R = extremely long period	~ 0.001	

Instrument code:

A = tiltmeter	B = creepmeter
D = pressure	F = magnetometer
G = gravimeter/accelerometer	H = high gain seismometer
I = humidity	K = temperature
L = low gain seismometer	M = mass position seismometer
P = geophone	R = rainfall
S = linear strainmeter	V = volumetric strain
T = tide meter	W = wind

Component code:

N = north-south	E = east-west
T = transverse	R = radial

Z = vertical
A, B, C = triaxial
1, 2, 3 = orthogonal components but non-traditional orientations
U, V, W = optional components

Name:	*selectflag*
Relations:	**Event**
Description:	This flag describes the “definitive” definition of an event. Helps to differentiate which event to use from two different real-time systems.
Datatype:	***int*** ***[number(1)]***
NULL Value:	***No***
Range:	*rflag* $\in$ {0 | 1} 0 = false 1 = true

Name:	<u>*slores*</u>
Relations:	**AssocArO**
Description:	Slowness residual. This attribute gives the difference between an observed slowness (saved in the **Arrival** relation) and a theorical prediction. The prediction is calculated for the related phase and event origin described in the record.
Datatype:	***float*** ***[number(8,4)]***
NULL Value:	***Yes***
Range:	*slores* $\geq$ 0.0
Units:	***Seconds/kilometer***

Name:	*slow*
Relations:	**Arrival**
Description:	Observed slowness. This is the observed slowness of a wave as it sweeps across an array.
Datatype:	***float [number(8,4)]***
NULL Value:	***Yes***
Range:	*slow* $\geq$ 0.0
Units:	***Seconds/kilometer***

Name:	*smxx, smyy, smzz, smxy, smxz, smyz*
Relations:	***Mec***
Description:	Uncertainties in moment tensor elements.
Datatype:	***float [number(4,2)]***
Range:	$-10^{32} <$ *smxx, smyy, smzz, smxy, smxz, smyz* $< 10^{32}$
Units:	***Dyne centimeters***

Name:	*snr*
Relations:	**Arrival**, **Amp**
Description:	Signal-to-noise ratio. This is an estimate of the signal relative to that of the noise immediately preceding it.
Datatype:	***float***
NULL Value:	***Yes***
Range:	*snr* > 0.0

Name: *srcduration*

Relations: **Mec**

Description: Source half-duration.

Datatype: ***float*** ***[number(6,3)]***

NULL Value: ***Yes***

Range: $0.0 < \textit{srcduration} < 100.0$

Units: ***Seconds***

Name: *sta*

Relations: **Arrival**, **Amp**, **Coda**

Description: Station code. This is the common code-name of a seismic observatory. Generally only three or four characters are used.

Datatype: ***char(6)***

NULL Value: ***No***

Range: Any free-format string up to 6 characters

Name *stime*

Relations: **Origin**

Description: Origin time error. This attribute denotes the standard deviation of the origin time.

Datatype: ***float*** ***[number(6,3)]***

NULL Value: ***Yes***

Range: $\textit{stime} \geq 0.0$

Units: ***Seconds***

Name:	*striken*
Relations:	**Mec**
Description:	Strike of null axis in the principal axes representation of the best double couple.
Datatype:	***int*** ***[number(3)]***
NULL Value:	***Yes***
Range:	0 ≤ strike*n* ≤ 360
Units:	***Degrees***

Name:	*strikep*
Relations:	**Mec**
Description:	Strike of compressional axis in the principal axes representation of the best double couple.
Datatype:	***int*** ***[number(3)]***
NULL Value:	***Yes***
Range:	0 ≤ strikep ≤ 360
Units:	***Degrees***

Name:	*striket*
Relations:	**Mec**
Description:	Strike of tension axis in the principal axes representation of the best double couple.
Datatype:	***int*** ***[number(3)]***
NULL Value:	***Yes***
Range:	0 ≤ striket ≤ 360
Units:	***Degrees***

Name:	*strike1*
Relations:	**Mec**
Description:	Strike of fault plane 1 from the best double couple.
Datatype:	***int*** ***[number(3)]***
NULL Value:	***Yes***
Range:	$0 \leq strike1 \leq 360$
Units:	***Degrees***

Name:	*strike2*
Relations:	**Mec**
Description:	Strike of fault plane 2 from the best double couple.
Datatype:	***int*** ***[number(3)]***
NULL Value:	***Yes***
Range:	$0 \leq strike2 \leq 360$
Units:	***Degrees***

Name:	*stx*, *sty*, *stz*, *sxx*, *sxy*, *sxz*, *syy*, *syz*, *stt*, *szz*
Relations:	**Origin_Error**
Description:	Elements of the covariance matrix for the location defined by *orid*. The covariance matrix is symmetric (and positive definite) so that sxy = syx, etc., (x, y, z, t) refer to latitude, longitude, depth and origin time, respectively. These attributes (together with *sdobs*, *ndef* and *dattype*) provide all the information necessary to construct the K-dimensional (K=2, 3, 4) confidence ellipse or ellipsoids at any confidence limit desired.
Datatype:	***float [number(5,3)]***
NULL Value:	***Yes***
Range:	*sxx*, *syy*, *szz*, *stt* > 0.0
Units:	*sxx*, *syy*, *szz*, *sxy*, *sxz*, *syz* – ***Kilometers squared*** *stt* – ***Seconds squared*** *stx*, *sty*, *stz* – ***Kilometers*second***

Name:	*subsource*
Relations:	**Event**, **Origin**, **Netmag**, **Arrival**, **Amp**, **Mec**, **Coda, AssocArO, AssocAmO, AssocAmM, AssocCoM, AssocCoO**
Description:	A second identifier to specify a unique source within the domain specified by *auth*.
Datatype:	***char(8)***
NULL Value:	***Yes***
Range:	Any free-format string up to 8 characters

Name:	*tau*
Relations:	**Amp**, **Coda**
Description:	Coda duration (F-P time)
Datatype:	***float*** ***[number(7,4)]***
NULL Value:	***Yes***
Range:	*tau* > 0.0
Units:	***Seconds***

Name:	*tfd*
Relations:	***Mec***
Description:	Duration of the source time function.
Datatype:	***float*** ***[number(6,3)]***
Range:	*tfd* > 0.0
Units:	***Seconds***

Name:	*tft*
Relations:	***Mec***
Description:	Type of source time function.
Datatype:	***char(8)***
Range:	Any free-format string up to 8 characters

Name:	*time1, time2, time3, time4, time5, time6*
Relations:	**Coda**
Description:	Time for pairs.
Datatype:	***int*** ***[number(10)]***
NULL Value:	***Yes***
Range:	*time1, time2, time3, time4, time5, time6* > 0
Units:	***Seconds***

Name:	*timeres*
Relations:	**AssocArO**
Description:	Time residual. This attribute is a travel time residual, measured in seconds. The residual is found by taking the observed arrival time (saved in the **Arrival** relation) of a seismic phase and substracting the expected arrival time. The expected arrival time is calculated by a formula based on earth velocity model, an event location and origin time, the distance to the station, and the particular seismic phase.
Datatype:	***float*** ***[number(5,2)]***
NULL Value:	***Yes***
Range:	*timeres* ≥ 0.0
Units:	***Seconds***

Name:	*totalamp*
Relations:	**Origin**
Description:	Total number of amplitudes in the database for the specified *orid*. Not all amplitudes have to associated with all magnitudes or origins.
Datatype:	***int*** ***[number(6)]***
NULL Value:	***No***
Range:	*totalamp* ≥ 0

Name:	*totalarr*
Relations:	**Origin**
Description:	Total number of phases in the database for the specified *orid*. This field signifies the number of P and S arrivals with non-zero input weights.
Datatype:	***int*** ***[number(5)]***
NULL Value:	***No***
Range:	*totalarr* ≥ 0

Name:	*type*
Relations:	**Origin**
Description:	Type of location.
Datatype:	***char(2)***
NULL Value:	***Yes***
Range:	*type* ∈ {H | C | A} H = hypocenter C = centroid A = amplitude

Name:	*uncertainty*
Relations:	**Netmag**
Description:	Magnitude uncertainty. This is the standard deviation of the accompanying magnitude measurement.
Datatype:	***float*** ***[number(5,3)]***
NULL Value:	***Yes***
Range:	*uncertainty* ≥ 0.0

Name:	*undip1*
Relations:	**Mec**
Description:	Uncertainty in dip 1.
Datatype:	***float [number(5,3)]***
NULL Value:	***Yes***
Range:	-180.0 ≤ *undip1* ≤ 180.0
Units:	***Degrees***

Name:	*undip2*
Relations:	**Mec**
Description:	Uncertainty in dip 2.
Datatype:	***float [number(5,3)]***
NULL Value:	***Yes***
Range:	-180.0 ≤ *undip2* ≤ 180.0
Units:	***Degrees***

Name:	*units*
Relations:	**Amp**, **Coda**
Description:	Units of amplitude.
Datatype:	***char(4)***
NULL Value:	***No***
Range:	*units* ∈ {c | s | mm | cm | m | ms | mss | cms | cmss | mms | mmss | mc | nm | e | iovs | spa } c = counts s = seconds mm = millimeters cm = centimeters m = meters ms = meters per second mss = meters per second per second cms = centimeters per second cmss = centimeters per second per second mms = millimeters per second mmss = millimeters per second per second mc = microns nm = nanometers e = ergs iovs = integral of velocity squared spa = spectral peak amplitude

Name:	*unrake1*
Relations:	**Mec**
Description:	Uncertainty in rake 1.
Datatype:	***float [number(6,3)]***
NULL Value:	***Yes***
Range:	-180.0 ≤ *unrake1* ≤ 180.0
Units:	***Degrees***

Name:	*unrake2*
Relations:	**Mec**
Description:	Uncertainty in rake 2.
Datatype:	***float [number(6,3)]***
NULL Value:	***Yes***
Range:	-180.0 ≤ *unrake2* ≤ 180.0
Units:	***Degrees***

Name:	*unstrike1*
Relations:	**Mec**
Description:	Uncertainty in strike 1.
Datatype:	***float [number(6,3)]***
NULL Value:	***Yes***
Range:	-180.0 ≤ *unstrike1* ≤ 180.0
Units:	***Degrees***

Name:	*unstrike2*
Relations:	**Mec**
Description:	Uncertainty in strike 2.
Datatype:	***float [number(6,3)]***
NULL Value:	***Yes***
Range:	-180.0 ≤ *unstrike2* ≤ 180.0
Units:	***Degrees***

Name:	*vmodelid*
Relations:	**Origin**
Description:	Velocity model identification.
Datatype:	***int*** ***[number(3)]***
NULL Value:	***Yes***
Range:	*vmodelid* > 0

Name:	*weight*
Relations:	**AssocAmM**, **AssocCoM**
Description:	Magnitude weight.
Datatype:	***float*** ***[number(4,3)]***
NULL Value:	***Yes***
Range:	0.0 < *weight* ≤ 1.0 0.0 = no weight (not used)

Name:	*wgt*
Relations:	**AssocArO**
Description:	Location weight. This attribute gives the final weight assigned to the allied arrival by the location program. It is used primarily for location programs that adaptively weight data by their residuals.
Datatype:	***float*** ***[number(4,3)]***
NULL Value:	***Yes***
Range:	0.0 ≤ *wgt* < 1.0

Name:	*wrms*
Relations:	**Origin**
Description:	This attribute denotes the weighted RMS.
Datatype:	***float [number(5,3)]***
NULL Value:	***Yes***
Range:	*wrms* > 0.0

Name:	*wstart*
Relations:	**Amp**
Description:	Start of time window of amplitude measurement.
Datatype:	***float***
NULL Value:	**No**
Range:	Any valid float. *wstart* = 0.0 on **January 01, 1970** at **00:00:00.0000**
Units:	***Seconds***

<u>5.- Primary keys :</u>

Oracle's sequences are used now to generate primary keys values instead of the **Lastid** table. A sequence is a database object from which multiple users may generate unique integers.

Values for a given sequence are automatically generated by special Oracle routines and, consequently, sequences avoid the performance bottleneck which results from implementation of sequences at the application level. For example, one common application-level implementation is to force each transaction to lock a sequence number table, increment the sequence, and then release the table. Under this implementation, only one sequence number may be generated at a time. In contrast, Oracle sequences permit the simultaneous generation of multiple sequence numbers while guaranteeing that every sequence number is unique.

When a sequence number is generated, the sequence is incremented, independent of the transaction committing or rolling back. If two users concurrently increment the same sequence, the sequence numbers each user acquires may have gaps because sequence numbers are being generated by the other user. One user can never acquire the sequence number generated by another user. Once a sequence value is generated by one user, that user can continue to access that value regardless of whether the sequence is incremented by another user.

Because sequence numbers are generated independently of tables, the same sequence can be used for one or for multiple tables. It is possible that individual sequence numbers will appear to be skipped, because they were generated and used in a transaction that ultimately rolled back. Additionally, a single user may not realize that other users are drawing from the same sequence.

For more information, see **Oracle8 Server SQL Reference Release 8.0**

6.- Time issues :

The *PL/SQL* package ***truetime*** is an implementation of the different functions to make the conversion between:

- String format time.
- Nominal epoch time.
- True epoch time (Atomic time TAI).

This package consists of six functions to convert between the different times. The origin time chosen to compute nominal and true epoch times is **January 01, 1970** at **00:00:00**.

To implement the concept of leap seconds, a table **leap_seconds** was needed. This table has the following structure :

Leap_seconds		Information about leap seconds
Attribute	***Type***	***Description***
s_nominal	int	Start nominal time
e_nominal	int	End nominal time
s_true	int	Start true time
e_true	int	End true time
ls_count	int	Leap seconds count

Each row of the **leap_seconds** table represents an interval between two leap seconds.

- *Function **string2nominal** :*

This function converts a string format time to a nominal epoch time.

Examples :

string2nominal ('1972/12/31 23:59:59') = 94694399
string2nominal ('1972/12/31 23:59:60') = NULL
string2nominal ('1972/12/31 23:59:61') = NULL
string2nominal ('1973/01/01 00:00:00') = 94694400

- *Function* ***nominal2string :***

This function converts a nominal epoch time to a string format time.

Examples :

nominal2string (0) = '1970/01/01 00:00:00'
nominal2string (94694399) = '1972/12/31 23:59:59'
nominal2string (94694400) = '1973/01/01 00:00:00'

- *Function* ***nominal2true :***

This function converts a nominal epoch time to a true epoch time.

Examples :

nominal2true (0) = 0
nominal2true (94694399) = 94694400
nominal2true (94694400) = 94694403

- *Function* ***true2nominal :***

This function converts a true epoch time to a nominal epoch time.

Examples :

true2nominal (94694400) = 94694399
true2nominal (94694401) = NULL
true2nominal (94694402) = NULL
true2nominal (94694403) = 94694400

- *Function* ***string2true :***

This function converts a string format time to a true epoch time.

Examples :

string2nominal ('1972/12/31 23:59:59') = 94694400
string2nominal ('1972/12/31 23:59:60') = 94694401
string2nominal ('1972/12/31 23:59:61') = 94694402
string2nominal ('1973/01/01 00:00:00') = 94694403

- *Function* ***true2string :***

This function converts a true epoch time to a string format time.

Examples :

true2string (94694400) = '1972/12/31 23:59:59'
true2string (94694401) = '1972/12/31 23:59:60'
true2string (94694402) = '1972/12/31 23:59:61'
true2string (94694403) = '1973/01/01 00:00:00'

The *PL/SQL* script ***Init_Leap_Seconds.sql*** populates the **leap_seconds** table.

```
--  /*******************************************************************
--  * Truetime.sql ---> PL/SQL Package for conversion between :        *
--  *                                                                   *
--  *       - String format time.                                       *
--  *       - Nominal epoch time.                                       *
--  *       - True epoch time.                                          *
--  *                                                                   *
--  * Stephane Zuzlewski @1998                                          *
--  *                                                                   *
--  *******************************************************************/

SET SERVEROUTPUT ON;

--  /*******************************************************************
--  * The table leap_seconds contains information about all the current *
--  * leap seconds.                                                     *
--  *******************************************************************/

DROP TABLE leap_seconds;

CREATE TABLE leap_seconds (
s_nominal   INTEGER,    /* Start nominal epoch time */
e_nominal   INTEGER,    /* End nominal epoch time   */
s_true      INTEGER,    /* Start true epoch time    */
e_true      INTEGER,    /* End true epoch time      */
ls_count    INTEGER     /* Leap seconds count       */
);


CREATE OR REPLACE PACKAGE truetime AS

      FUNCTION string2nominal (d VARCHAR) RETURN INTEGER;
      FUNCTION nominal2string (n INTEGER) RETURN VARCHAR;
      FUNCTION nominal2true   (n INTEGER) RETURN INTEGER;
      FUNCTION true2nominal   (t INTEGER) RETURN INTEGER;
      FUNCTION string2true    (d VARCHAR) RETURN INTEGER;
      FUNCTION true2string    (t INTEGER) RETURN VARCHAR;

      PRAGMA RESTRICT_REFERENCES (string2nominal, WNDS);
      PRAGMA RESTRICT_REFERENCES (nominal2string, WNDS);
      PRAGMA RESTRICT_REFERENCES (nominal2true,   WNDS);
      PRAGMA RESTRICT_REFERENCES (true2nominal,   WNDS);
      PRAGMA RESTRICT_REFERENCES (string2true,    WNDS);
      PRAGMA RESTRICT_REFERENCES (true2string,    WNDS);
END truetime;
/

SHOW ERRORS PACKAGE truetime;
```

```
CREATE OR REPLACE PACKAGE BODY truetime AS

--  /*****************************************************************
--  * The function string2nominal converts a string format time to a      *
--  * nominal epoch time.                                                   *
--  *                                                                       *
--  * Returns NULL if the time is a leap second.                            *
--  *****************************************************************/

FUNCTION string2nominal (d VARCHAR) RETURN INTEGER IS
      origin      CONSTANT DATE := '1970/01/01 00:00:00';
      origin_days INTEGER;
      d_days      INTEGER;
      diff_days   INTEGER;
      nominal     INTEGER;
      d2          VARCHAR(20);
      d3          VARCHAR(20);
BEGIN
      d2          := REPLACE (d, ':60', ':59');
      d3          := REPLACE (d, ':61', ':59');

      IF    d != d2 THEN
            RETURN (NULL);
      ELSIF d != d3 THEN
            RETURN (NULL);
      ELSE
            origin_days := TO_CHAR (origin, 'J');
            d_days      := TO_CHAR (TO_DATE (d, 'YYYY/MM/DD HH24:MI:SS'),
'J');

            diff_days   := d_days - origin_days;

            nominal     := (diff_days*86400) + TO_CHAR (TO_DATE (d,
'YYYY/MM/DD HH24:MI:SS'), 'SSSSS');

            RETURN (nominal);
      END IF;
END string2nominal;

--  /*****************************************************************
--  * The function nominal2string converts a nominal epoch time to a      *
--  * string format time.                                                   *
--  *****************************************************************/

FUNCTION nominal2string (n INTEGER) RETURN VARCHAR IS
      origin      CONSTANT DATE := '1970/01/01 00:00:00';
      d           VARCHAR(20);
BEGIN
      d           := TO_CHAR (TO_DATE (origin, 'YYYY/MM/DD HH24:MI:SS') +
(n/86400), 'YYYY/MM/DD HH24:MI:SS');

      RETURN (d);
END nominal2string;
```

```
--  /*******************************************************************
--  * The function nominal2true converts a nominal epoch time to a true    *
--  * epoch time.                                                          *
--  ********************************************************************/

FUNCTION nominal2true (n INTEGER) RETURN INTEGER IS
      true        INTEGER;
      lp_secs     INTEGER;
BEGIN
      SELECT ls_count INTO lp_secs FROM leap_seconds
      WHERE s_nominal <= n
      AND   e_nominal >= n;

      true        := n + lp_secs;

      RETURN (true);
END nominal2true;

--  /*******************************************************************
--  * The function true2nominal converts a true epoch time to a nominal    *
--  * epoch time.                                                          *
--  *                                                                      *
--  * Returns NULL if the time is a leap second.                           *
--  ********************************************************************/

FUNCTION true2nominal (t INTEGER) RETURN INTEGER IS
      nominal           INTEGER;
BEGIN
      nominal           := string2nominal (true2string (t));

      RETURN (nominal);
END true2nominal;

--  /*******************************************************************
--  * The function string2true converts a string format time to a true     *
--  * epoch time.                                                          *
--  ********************************************************************/

FUNCTION string2true (d VARCHAR) RETURN INTEGER IS
      true        INTEGER;
      leap        INTEGER;
      d2          VARCHAR(20);
      d3          VARCHAR(20);
      d4          VARCHAR(20);
BEGIN
      d2          := REPLACE (d, ':60', ':59');
      d3          := REPLACE (d, ':61', ':59');

      IF    d != d2 THEN
            d4    := d2;
            leap  := 1;
      ELSIF d != d3 THEN
            d4    := d3;
            leap  := 2;
      ELSE
            d4    := d;
            leap  := 0;
      END IF;
```

```
      true        := nominal2true (string2nominal (d4)) + leap;

      RETURN (true);
END string2true;

--  /*****************************************************************
--  * The function true2string converts a true epoch time to a string      *
--  * format time.                                                          *
--  *****************************************************************/

FUNCTION true2string (t INTEGER) RETURN VARCHAR IS
      d           VARCHAR(20);
      lp_secs1    INTEGER;
      lp_secs2    INTEGER;
      lp_secs3    INTEGER;
BEGIN

      BEGIN
      SELECT ls_count INTO lp_secs1 FROM leap_seconds
      WHERE s_true <= t
      AND   e_true >= t;
      EXCEPTION WHEN NO_DATA_FOUND THEN lp_secs1 := -1;
      END;

      IF    lp_secs1 >= 0     THEN
            d     := nominal2string (t - lp_secs1);
      ELSE
            BEGIN
            SELECT ls_count INTO lp_secs2 FROM leap_seconds
            WHERE s_true <= t-1
            AND   e_true >= t-1;
            EXCEPTION WHEN NO_DATA_FOUND THEN lp_secs2 := -1;
            END;

            IF    lp_secs2 >= 0     THEN
                  d     := nominal2string (t - lp_secs2 - 2);
                  d     := REPLACE (d, ':58', ':60');
            ELSE
                  BEGIN
                  SELECT ls_count INTO lp_secs3 FROM leap_seconds
                  WHERE s_true <= t-2
                  AND   e_true >= t-2;
                  END;

                  d     := nominal2string (t - lp_secs3 - 3);
                  d     := REPLACE (d, ':58', ':61');
            END IF;
      END IF;


      RETURN (d);
END true2string;

END truetime;
/

SHOW ERRORS PACKAGE BODY truetime;

COMMIT;
EXIT;
```

```
--  /*****************************************************************
--  * Init_Leap_Secs.sql ---> PL/SQL Script for initialization of the      *
--  *                         Leap seconds table.                          *
--  *                                                                      *
--  * Stephane Zuzlewski @1998                                             *
--  *                                                                      *
--  *****************************************************************/


-- /*****************************************************************
-- * The table leap_seconds_tmp is a temporary table needed to populate   *
-- * the leap_seconds table.                                              *
-- *****************************************************************/

DROP TABLE leap_seconds_tmp;

CREATE TABLE leap_seconds_tmp (
s_nominal   INTEGER,    /* Start nominal epoch time */
e_nominal   INTEGER,    /* End nominal epoch time   */
s_true      INTEGER,    /* Start true epoch time    */
e_true      INTEGER,    /* End true epoch time      */
ls_count    INTEGER     /* Leap seconds count       */
);

-- /*****************************************************************
-- * Deleting all the rows in the leap_seconds table                      *
-- *****************************************************************/

DELETE FROM leap_seconds;


--  /************************************
--  * Populating the leap_seconds table *
--  ************************************/

DECLARE

--  /* Leap seconds in string format time */

l0    DATE := '0001/01/01 00:00:00'; -- /* Represents minus infinite */
l1    DATE := '1972/06/30 23:59:59';
l2    DATE := '1972/12/31 23:59:59';
l3    DATE := '1973/12/31 23:59:59';
l4    DATE := '1974/12/31 23:59:59';
l5    DATE := '1975/12/31 23:59:59';
l6    DATE := '1976/12/31 23:59:59';
l7    DATE := '1977/12/31 23:59:59';
l8    DATE := '1978/12/31 23:59:59';
l9    DATE := '1979/12/31 23:59:59';
l10   DATE := '1981/06/30 23:59:59';
l11   DATE := '1982/06/30 23:59:59';
l12   DATE := '1983/06/30 23:59:59';
l13   DATE := '1985/06/30 23:59:59';
l14   DATE := '1987/12/31 23:59:59';
l15   DATE := '1989/12/31 23:59:59';
l16   DATE := '1990/12/31 23:59:59';
l17   DATE := '1992/06/30 23:59:59';
l18   DATE := '1993/06/30 23:59:59';
l19   DATE := '1994/06/30 23:59:59';
```

```
l20   DATE := '1995/12/31 23:59:59';
l21   DATE := '1997/06/30 23:59:59';
l22   DATE := '3000/01/01 00:00:00'; -- /* Represents plus infinite */

--  /* Leap seconds in nominal epoch time */
n0    INTEGER;
n1    INTEGER;
n2    INTEGER;
n3    INTEGER;
n4    INTEGER;
n5    INTEGER;
n6    INTEGER;
n7    INTEGER;
n8    INTEGER;
n9    INTEGER;
n10   INTEGER;
n11   INTEGER;
n12   INTEGER;
n13   INTEGER;
n14   INTEGER;
n15   INTEGER;
n16   INTEGER;
n17   INTEGER;
n18   INTEGER;
n19   INTEGER;
n20   INTEGER;
n21   INTEGER;
n22   INTEGER;

--  /* Leap seconds in true epoch time */
t0    INTEGER;
t1    INTEGER;
t2    INTEGER;
t3    INTEGER;
t4    INTEGER;
t5    INTEGER;
t6    INTEGER;
t7    INTEGER;
t8    INTEGER;
t9    INTEGER;
t10   INTEGER;
t11   INTEGER;
t12   INTEGER;
t13   INTEGER;
t14   INTEGER;
t15   INTEGER;
t16   INTEGER;
t17   INTEGER;
t18   INTEGER;
t19   INTEGER;
t20   INTEGER;
t21   INTEGER;
t22   INTEGER;

tt1   INTEGER;
tt2   INTEGER;
tt3   INTEGER;
tt4   INTEGER;
tt5   INTEGER;
tt6   INTEGER;
```

```
tt7   INTEGER;
tt8   INTEGER;
tt9   INTEGER;
tt10  INTEGER;
tt11  INTEGER;
tt12  INTEGER;
tt13  INTEGER;
tt14  INTEGER;
tt15  INTEGER;
tt16  INTEGER;
tt17  INTEGER;
tt18  INTEGER;
tt19  INTEGER;
tt20  INTEGER;
tt21  INTEGER;

BEGIN
--      /* Computing nominal epoch times */
        n0  := truetime.string2nominal (l0);
        n1  := truetime.string2nominal (l1);
        n2  := truetime.string2nominal (l2);
        n3  := truetime.string2nominal (l3);
        n4  := truetime.string2nominal (l4);
        n5  := truetime.string2nominal (l5);
        n6  := truetime.string2nominal (l6);
        n7  := truetime.string2nominal (l7);
        n8  := truetime.string2nominal (l8);
        n9  := truetime.string2nominal (l9);
        n10 := truetime.string2nominal (l10);
        n11 := truetime.string2nominal (l11);
        n12 := truetime.string2nominal (l12);
        n13 := truetime.string2nominal (l13);
        n14 := truetime.string2nominal (l14);
        n15 := truetime.string2nominal (l15);
        n16 := truetime.string2nominal (l16);
        n17 := truetime.string2nominal (l17);
        n18 := truetime.string2nominal (l18);
        n19 := truetime.string2nominal (l19);
        n20 := truetime.string2nominal (l20);
        n21 := truetime.string2nominal (l21);
        n22 := truetime.string2nominal (l22);


--      /* Populating the leap_seconds table */
        INSERT INTO leap_seconds VALUES (n0,    n1,  NULL, NULL, 0);
        INSERT INTO leap_seconds VALUES (n1+1,  n2,  NULL, NULL, 1);
        INSERT INTO leap_seconds VALUES (n2+1,  n3,  NULL, NULL, 2);
        INSERT INTO leap_seconds VALUES (n3+1,  n4,  NULL, NULL, 3);
        INSERT INTO leap_seconds VALUES (n4+1,  n5,  NULL, NULL, 4);
        INSERT INTO leap_seconds VALUES (n5+1,  n6,  NULL, NULL, 5);
        INSERT INTO leap_seconds VALUES (n6+1,  n7,  NULL, NULL, 6);
        INSERT INTO leap_seconds VALUES (n7+1,  n8,  NULL, NULL, 7);
        INSERT INTO leap_seconds VALUES (n8+1,  n9,  NULL, NULL, 8);
        INSERT INTO leap_seconds VALUES (n9+1,  n10, NULL, NULL, 9);
        INSERT INTO leap_seconds VALUES (n10+1, n11, NULL, NULL, 10);
        INSERT INTO leap_seconds VALUES (n11+1, n12, NULL, NULL, 11);
        INSERT INTO leap_seconds VALUES (n12+1, n13, NULL, NULL, 12);
        INSERT INTO leap_seconds VALUES (n13+1, n14, NULL, NULL, 13);
        INSERT INTO leap_seconds VALUES (n14+1, n15, NULL, NULL, 14);
        INSERT INTO leap_seconds VALUES (n15+1, n16, NULL, NULL, 15);
```

```
        INSERT INTO leap_seconds VALUES (n16+1, n17, NULL, NULL, 16);
        INSERT INTO leap_seconds VALUES (n17+1, n18, NULL, NULL, 17);
        INSERT INTO leap_seconds VALUES (n18+1, n19, NULL, NULL, 18);
        INSERT INTO leap_seconds VALUES (n19+1, n20, NULL, NULL, 19);
        INSERT INTO leap_seconds VALUES (n20+1, n21, NULL, NULL, 20);
        INSERT INTO leap_seconds VALUES (n21+1, n22, NULL, NULL, 21);


--      /* Computing true epoch times */
        t0  := truetime.nominal2true (n0);
        t1  := truetime.nominal2true (n1);
        t2  := truetime.nominal2true (n2);
        t3  := truetime.nominal2true (n3);
        t4  := truetime.nominal2true (n4);
        t5  := truetime.nominal2true (n5);
        t6  := truetime.nominal2true (n6);
        t7  := truetime.nominal2true (n7);
        t8  := truetime.nominal2true (n8);
        t9  := truetime.nominal2true (n9);
        t10 := truetime.nominal2true (n10);
        t11 := truetime.nominal2true (n11);
        t12 := truetime.nominal2true (n12);
        t13 := truetime.nominal2true (n13);
        t14 := truetime.nominal2true (n14);
        t15 := truetime.nominal2true (n15);
        t16 := truetime.nominal2true (n16);
        t17 := truetime.nominal2true (n17);
        t18 := truetime.nominal2true (n18);
        t19 := truetime.nominal2true (n19);
        t20 := truetime.nominal2true (n20);
        t21 := truetime.nominal2true (n21);
        t22 := truetime.nominal2true (n22);

        tt1       := truetime.nominal2true (n1+1);
        tt2       := truetime.nominal2true (n2+1);
        tt3       := truetime.nominal2true (n3+1);
        tt4       := truetime.nominal2true (n4+1);
        tt5       := truetime.nominal2true (n5+1);
        tt6       := truetime.nominal2true (n6+1);
        tt7       := truetime.nominal2true (n7+1);
        tt8       := truetime.nominal2true (n8+1);
        tt9       := truetime.nominal2true (n9+1);
        tt10      := truetime.nominal2true (n10+1);
        tt11      := truetime.nominal2true (n11+1);
        tt12      := truetime.nominal2true (n12+1);
        tt13      := truetime.nominal2true (n13+1);
        tt14      := truetime.nominal2true (n14+1);
        tt15      := truetime.nominal2true (n15+1);
        tt16      := truetime.nominal2true (n16+1);
        tt17      := truetime.nominal2true (n17+1);
        tt18      := truetime.nominal2true (n18+1);
        tt19      := truetime.nominal2true (n19+1);
        tt20      := truetime.nominal2true (n20+1);
        tt21      := truetime.nominal2true (n21+1);




        /* Populating the leap_seconds_tmp table */
```

```
        INSERT INTO leap_seconds_tmp VALUES (n0,    n1,  t0,   t1,   0);
        INSERT INTO leap_seconds_tmp VALUES (n1+1,  n2,  tt1,  t2,   1);
        INSERT INTO leap_seconds_tmp VALUES (n2+1,  n3,  tt2,  t3,   2);
        INSERT INTO leap_seconds_tmp VALUES (n3+1,  n4,  tt3,  t4,   3);
        INSERT INTO leap_seconds_tmp VALUES (n4+1,  n5,  tt4,  t5,   4);
        INSERT INTO leap_seconds_tmp VALUES (n5+1,  n6,  tt5,  t6,   5);
        INSERT INTO leap_seconds_tmp VALUES (n6+1,  n7,  tt6,  t7,   6);
        INSERT INTO leap_seconds_tmp VALUES (n7+1,  n8,  tt7,  t8,   7);
        INSERT INTO leap_seconds_tmp VALUES (n8+1,  n9,  tt8,  t9,   8);
        INSERT INTO leap_seconds_tmp VALUES (n9+1,  n10, tt9,  t10,  9);
        INSERT INTO leap_seconds_tmp VALUES (n10+1, n11, tt10, t11, 10);
        INSERT INTO leap_seconds_tmp VALUES (n11+1, n12, tt11, t12, 11);
        INSERT INTO leap_seconds_tmp VALUES (n12+1, n13, tt12, t13, 12);
        INSERT INTO leap_seconds_tmp VALUES (n13+1, n14, tt13, t14, 13);
        INSERT INTO leap_seconds_tmp VALUES (n14+1, n15, tt14, t15, 14);
        INSERT INTO leap_seconds_tmp VALUES (n15+1, n16, tt15, t16, 15);
        INSERT INTO leap_seconds_tmp VALUES (n16+1, n17, tt16, t17, 16);
        INSERT INTO leap_seconds_tmp VALUES (n17+1, n18, tt17, t18, 17);
        INSERT INTO leap_seconds_tmp VALUES (n18+1, n19, tt18, t19, 18);
        INSERT INTO leap_seconds_tmp VALUES (n19+1, n20, tt19, t20, 19);
        INSERT INTO leap_seconds_tmp VALUES (n20+1, n21, tt20, t21, 20);
        INSERT INTO leap_seconds_tmp VALUES (n21+1, n22, tt21, t22, 21);

END;
/


--  /* Destroying the leap_seconds table */
DROP TABLE leap_seconds;

-- /* Creating the leap_seconds table as a copy of the leap_seconds_tmp table
*/
CREATE TABLE leap_seconds AS SELECT * FROM leap_seconds_tmp;

--  /* Destroying the leap_seconds_tmp table */
DROP TABLE leap_seconds_tmp;

COMMIT;
EXIT;
```