

Modelul Entitate-Relație

Case*Method: Entity Relationship Modelling

Richard Barker - Oracle

Graphical Elements of ER Diagram

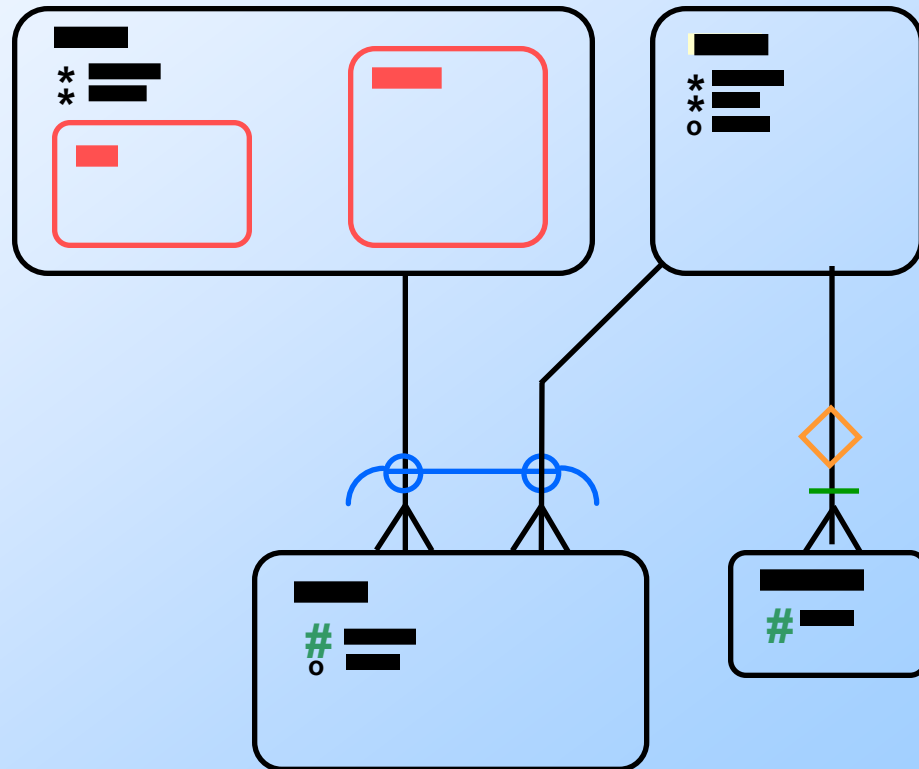
- ✓ Entity
- ✓ Attribute
- ✓ Relationship

Subtype

Unique identifier

Arc

Nontransferability



Exemplul de bază (ipotetic)

La călătoria cu avionul cu o linie aeriană un concept foarte tangibil este biletul de călătorie, ce poate acoperi un zbor să zicem din Atlantia (capitala grupului de insule Atlantis) la Paris.

Biletul este compus din cupoane, fiecare din ele folosind la un zbor între două aeroporturi.

Presupunem că biletul ipotetic este compus din două cupoane unul din Atlantia la Londra și altul de la Londra la Paris.

Airline: Atlantis Island Flights
Passenger:

Origin / Destination: Atlantia
Date of issue: 3 October 2008

		CARRIER	FLIGHT	CLASS	DATE	TIME	STATUS	COMMENT
From	Atlantia	AA	AIF	213	N	15/10/08	08:00	OK
To	London	HR	BA	424	N	16/10/08	21:30	OK
To	Paris	CD						

Fare: US\$ 845

Fiecare avion face mai multe zboruri în fiecare zi, identificat de data și ora plecării, numărul de zbor și aeroportul de plecare.

Numărul de zbor constă din două bucăți de informație: prima ne spune linia de zbor, în cazul de mai sus AIF (Atlantis Island Flights); a doua parte identifică unic ruta de zbor.

De aceea este nevoie să se cunoască ce avion a fost desemnat fiecărui zbor, câte bilete au fost vândute, dacă zborurile au fost confirmate și ce locuri au fost alocate.

Nucleul acestui sistem simplu este cuponul.

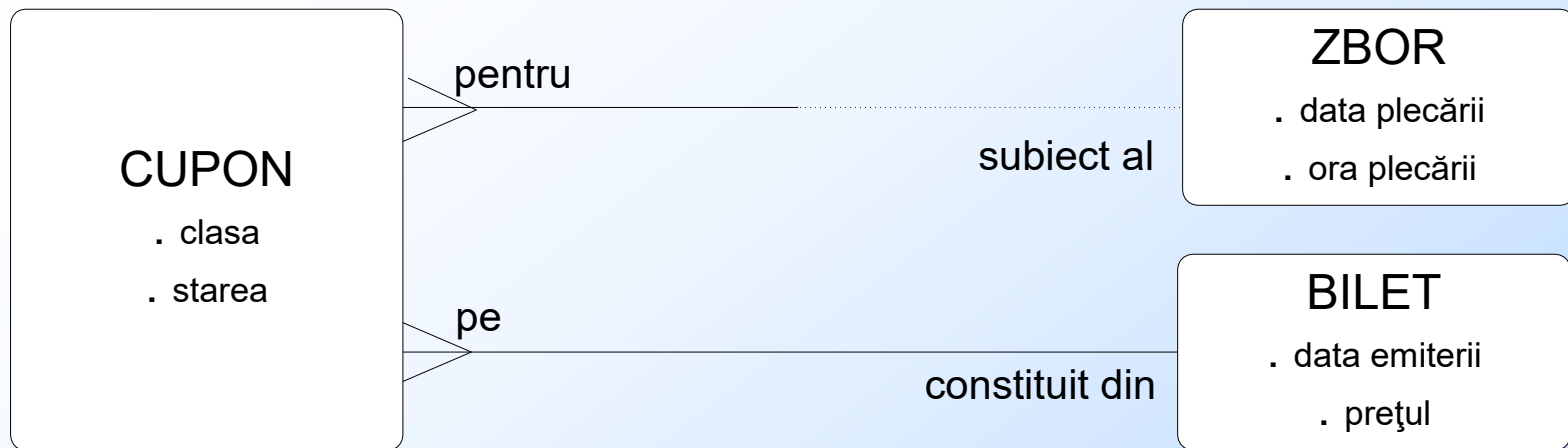
Este semnificativ ca denominatorul comun cel mai de jos și conține informații vitale ca de exemplu clasa ("class") și starea ("status").

El poate exista numai în contextul unui bilet, de la care moștenește data emiterii ("date of issue") și prețul ("fare").



Fiecare CUPON trebuie să fie pe un BILET și numai unul și
fiecare BILET trebuie să fie constituit din unul sau mai multe
CUPOANE

- Fiecare dreptunghi din diagrama precedentă reprezintă o entitate și linia între ele este o relație de legătură.
- Linia are o ramificație ("many") la capătul din stânga și o terminație simplă ("one") la capătul din dreapta, indicând faptul că pot exista mai multe cupoane pentru un bilet.
- Este o relație de legătură "many-to-one".
- Linia este continuă pentru a indica faptul că relația de legătură este obligatorie ("mandatory") și reprezintă o constrângere.



Fiecare CUPON trebuie să fie *pentru* un ZBOR și numai unul.

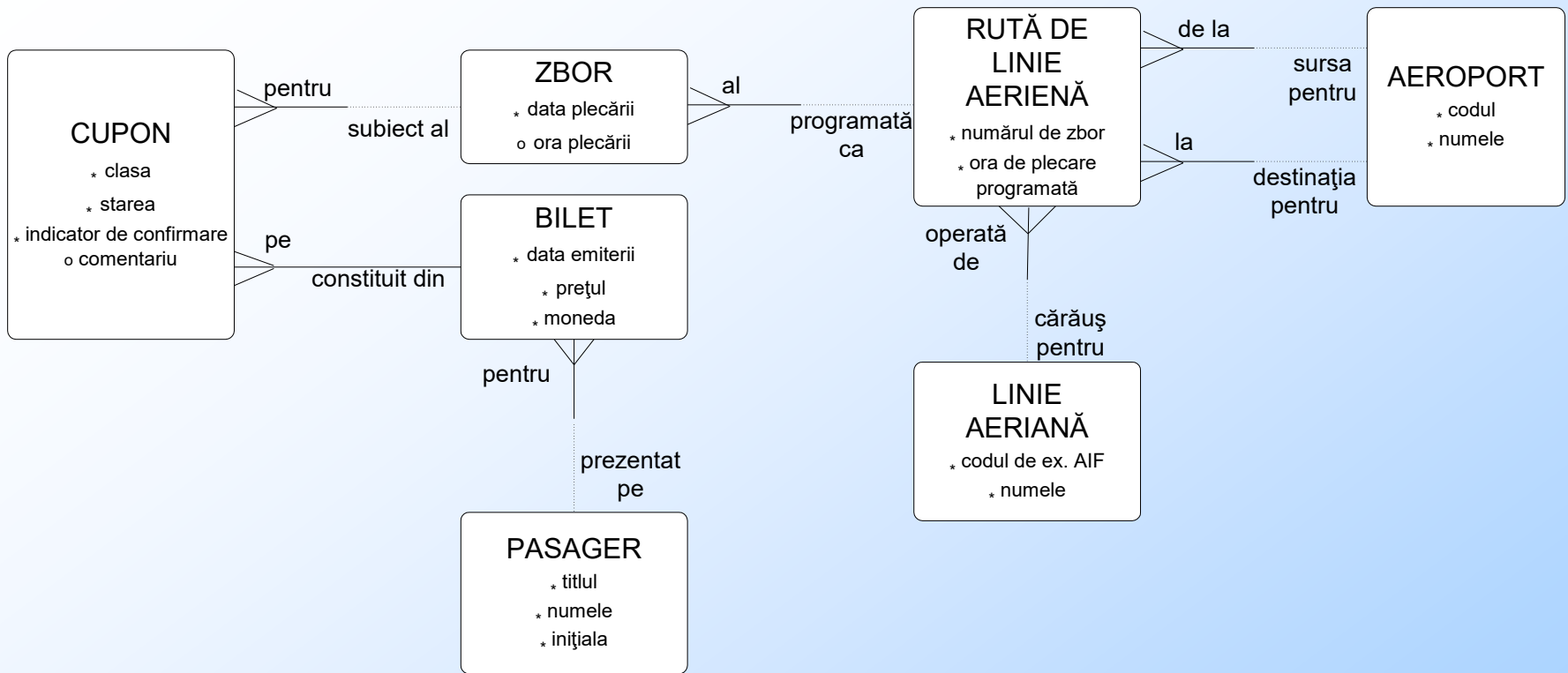
Fiecare ZBOR poate fi *subiect al* unuia sau mai multor CUPOANE.

- De notat că zborul poate să nu fie subiect al nici unui cupon.
- Acest lucru este arătat prin partea punctată a liniei, și înseamnă că relația de legătură este opțională.
- Se poate întâmpla să fie așa când tocmai s-a introdus un nou zbor și încă nu s-au vândut bilete (cuponul nu poate exista decât în contextul unui bilet).

Fiecare BILET trebuie să fie *constituit din* unul sau mai multe CUPOANE, fiecare din ele trebuie să fie *pentru* un ZBOR și numai unul.

Fiecare ZBOR poate fi *subiect al* unuia sau mai multor CUPOANE *fiecare* din ele trebuie să fie *pe* un BILET și numai unul.

- Avem acum o relație de legătură puternică între bilet și zbor, care de fapt este "many-to-many" (m:n).



- ❑ În interiorul dreptunghiurilor sunt prezentate alte informații utile.
- ❑ Aceste cuvinte reprezintă atribute ce descriu în detaliu entitățile.
- ❑ Fiecare BILET are data emiterii și preț.
- ❑ Ruta unei linii aeriene este identificată unic de numărul de zbor și linia aeriană asociată.
- ❑ Acest lucru este arătat pe bilet de un cod structurat ca de exemplu AIF213 sau AIF217.

- Fiecare LINIE AERIANĂ poate fi cărauş pentru una sau mai multe RUTE DE LINIE AERIANĂ, fiecare din care poate fi programată ca unul sau mai multe zboruri descrise de data şi ora de plecare.
- Conform diagramei fiecare ZBOR trebuie să fie al unei RUTE DE LINIE AERIANĂ predefinite (zboruri nestandard nu sunt permise) şi trebuie să se facă de la un AEROPORT la alt AEROPORT.
- Propoziţia de mai sus, care a fost citită de pe diagrama modelului ER, poate fi văzută din două unghiuri diferite.

Unghiul 1

Ce zboruri sunt programate de pe aeroportul Atlantia de către orice linie aeriană?
Acesta poate fi folosit pentru a afișa lista plecărilor.

PLECĂRI din AEROPORT				
Aeroport: Atlantia				
Linie aeriană	Numărul zborului	PLECARE		Destinație
		Data	Ora	
BA	962	3 Oct 2008	07:30	Londra
AIF	213	3 Oct 2008	08:00	Londra
AIF	004	3 Oct 2008	08:15	NewYork
BA	964	3 Oct 2008	09:00	Manchester
TW	051	3 Oct 2008	09:15	Chicago

Unghiul 2

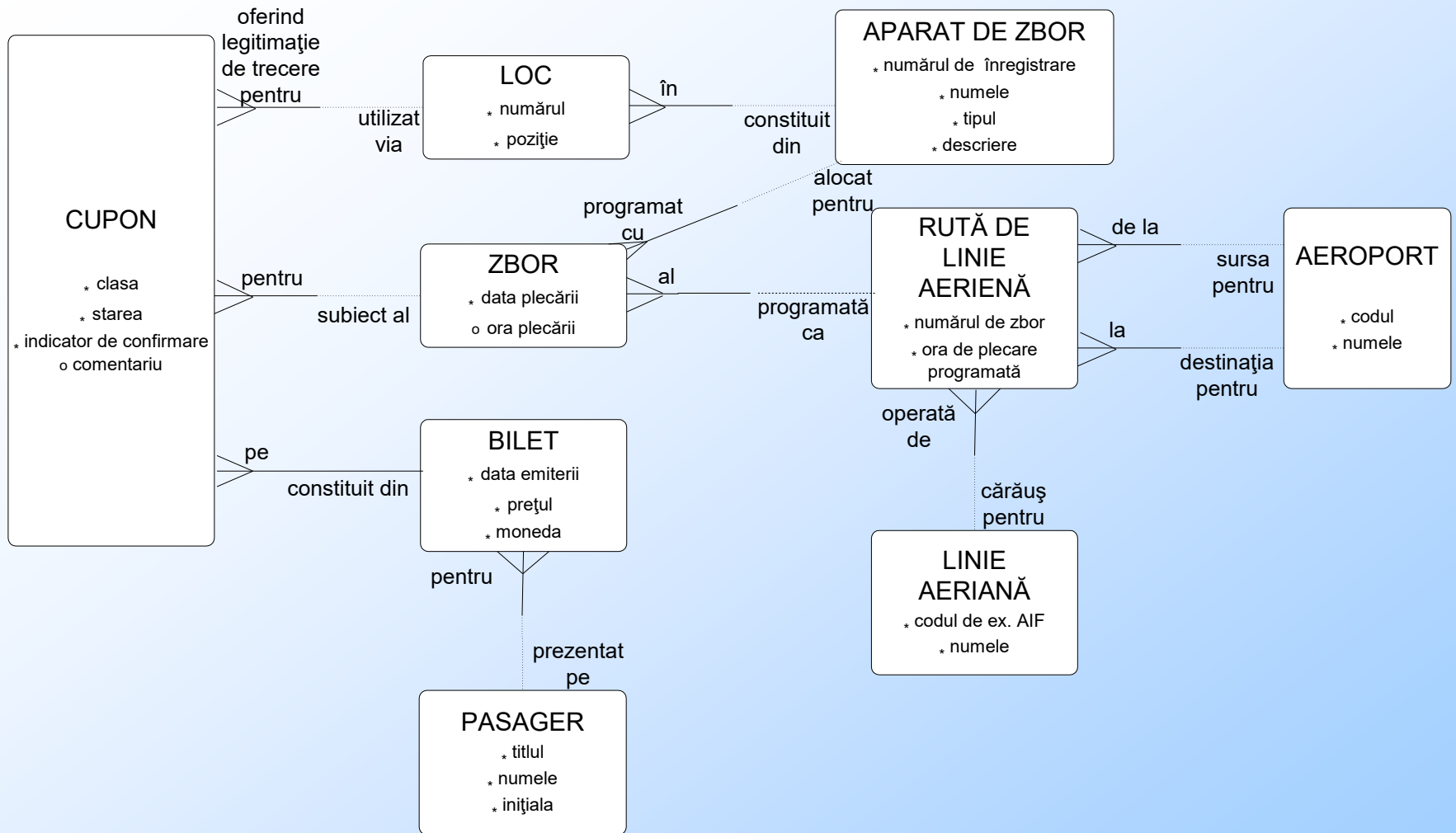
Ce zboruri au fost programate pentru Atlantis Island Flights din Atlantia?

Acesta va putea fi folosit de AIF pentru a păstra istoricul propriilor zboruri.

PLECĂRI pentru LINIE AERIANĂ			
Linia aeriană: Atlantis Island Flights (AIF)			
Aeroport: Atlantia			
Numărul zborului	Către	Plecare	
		Data	Ora
213	Londra LHR	3 Oct 2008	08:00
004	NewYork JFK	3 Oct 2008	08:15
009	South Island AASI	3 Oct 2008	09:20

- Cu alte cuvinte, liniile aeriene vor avea o bază de date ce poate fi folosită pentru a înregistra, raporta și controla informația modelată de diagrama ER.
- Cum se ajunge la forma finală acceptată a diagramei ER?
- De fiecare dată când informația de pe bilet sau cupon s-a referit la ceva din lumea reală, s-a creat o entitate (reprezentată printr-un dreptunghi cu numele scris cu majuscule) și apoi s-au pus datele recepționate ca attribute a celui mai apropiat lucru.
- Un atribut este orice descriere a unei entități.
- De exemplu entitatea LINIE AERIANĂ are un atribut intitulat *cod*.

- Oricând am găsit o asociere semnificativă între două entități, am adăugat o linie cu cuvinte de ambele părți (capete) pentru a reprezenta relații de legătură.
- În lumea reală codurile și numele sunt adesea folosite pentru a reprezenta relațiile.
- De exemplu în imaginea cuponului de pe un bilet codul AIF a fost utilizat în coloana intitulată "carrier" (cărăuș) pentru a semnifica o legătură cu Atlantis Island Flights (linie aeriană).
- Pe diagramă această relație a fost arătată indirect de liniile de la CUPON la ZBOR și la RUTĂ DE LINIE AERIANĂ și prin aceasta la LINIE AERIANĂ, și AIF este pur și simplu o valoare a atributului cod a entității LINIE AERIANĂ.



Modelul extins

Modelul extins

- Fiecare APARAT DE ZBOR poate fi *alocat pentru unul sau mai multe* ZBORURI (cu o dată și oră de plecare) și fiecare ZBOR trebuie să fie *a/ unei* RUTE (standard) DE LINIE AERIANĂ *de la un* AEROPORT *la alt* AEROPORT.
- Modelul permite chiar o rută de la un aeroport la el însuși.
- Fiecare APARAT DE ZBOR poate fi *constituit din unul sau mai multe* LOCURI, fiecare din ele putând fi (în mod fericit) *utilizat via* unul sau mai multe CUPOANE (sau PERMISE DE TRECERE) *pentru un* ZBOR (cu o dată și oră de plecare anume).

Modelul extins

- Fiecare BILET trebuie să fie pentru un PASAGER și numai unul, ce poate fi prezent pe unul sau mai multe BILETE.
- PASAGER este o entitate folositoare ce permite să se extindă ulterior modelul, pentru a ajuta linia aeriană să își personalizeze serviciul acordând fiecărui pasager locul preferat.
- O aplicație bună va asigura unicitatea cuponului (legitimației de trecere) pentru un loc la un zbor anume, deși înainte de alocarea locului un anumit nivel de "overbooking" este normal.
- De pe cupon (legitimația de trecere) se poate identifica evident lista pasagerilor avuți la bord, lucru ce poate fi foarte folositor în caz de urgențe sau dacă un pasager nu s-a urcat încă la bord cu câteva minute înainte de plecarea avionului.

Schema bazei de date:

Tabela	Coloana
APARAT DE ZBOR	Nume char(40) not null
	Număr de înregistrare char(20) not null
	Tip char(6) not null
	Descriere char(40) not null
LOC	Număr char(3) not null
	Număr de înregistrare al avionului char(20) not null **
	Poziție char(6) not null
CUPON	Codul liniei aeriene char(4) not null **
	Numărul zborului integer(4) not null **
	Data plecării date not null **
	Data emiterii date not null **
	Titlul pasagerului char(3) not null **
	Numele pasagerului char(30) not null **
	Inițiala pasagerului char(1) not null **
	Clasa char(1) not null
	Starea char(2) not null
	Indicator de confirmare char(1) not null
	Comentariu char(40) null
	Numărul locului char(3) null **

Oracle Data Types

□ Most Commonly-Used Oracle Data Types

- 1. CHAR(size)** These are fixed-length character data of length-sized bytes. Maximum size is 2000 bytes.
 - **Typical use:** for official International Currency Codes which are a fixed three characters in length such as USD, FFR.
- 2. VARCHAR2(size)** Variable-length character string having maximum length-sized bytes. Maximum size is 4000, and minimum is 1. This is the most commonly-used data type and you should use it if you are not sure which one to use. It replaces the old Oracle version 6 CHAR data type.
 - **Typical use:** for storing individual ASCII text *lines* of unlimited length ASCII texts on which you need to be able to search using a wildcard.
- 3. NUMBER** This data type is used for numerical values, with or without a decimal, of virtually unlimited size. Use this data type for data on which calculation or sorting should be possible. Avoid its use for numbers like a phone number, where the value does not have any meaning.
 - **Typical use:** amount of money, quantities, generated unique key values.

Oracle Data Types

□ Most Commonly-Used Oracle Data Types

4. **DATE** Valid date range from January 1, 4712 BC to December 31, 4712 AD. A date data type also contains time components. You should use it only when you know the full date including day, month, and year. The time component is often set to 00:00 (midnight) in normal use of dates.
 - **Typical use:** any date where the full date is known.
5. **LONG** Character data of variable length up to 2 gigabytes. Obsolete since Oracle8. Was used for ASCII text files where you do not need to search using the wildcard or substring functionality. Use CLOB data type instead.
 - **Typical use:** for storing the source code of HTML pages.
6. **LONG RAW** Raw binary data of variable length up to 2 gigabytes. Obsolete since Oracle8. Was used for large object types where the database should not try to interpret the data. Use BLOB data type instead.
 - **Typical use:** images or video clips.

Oracle Data Types

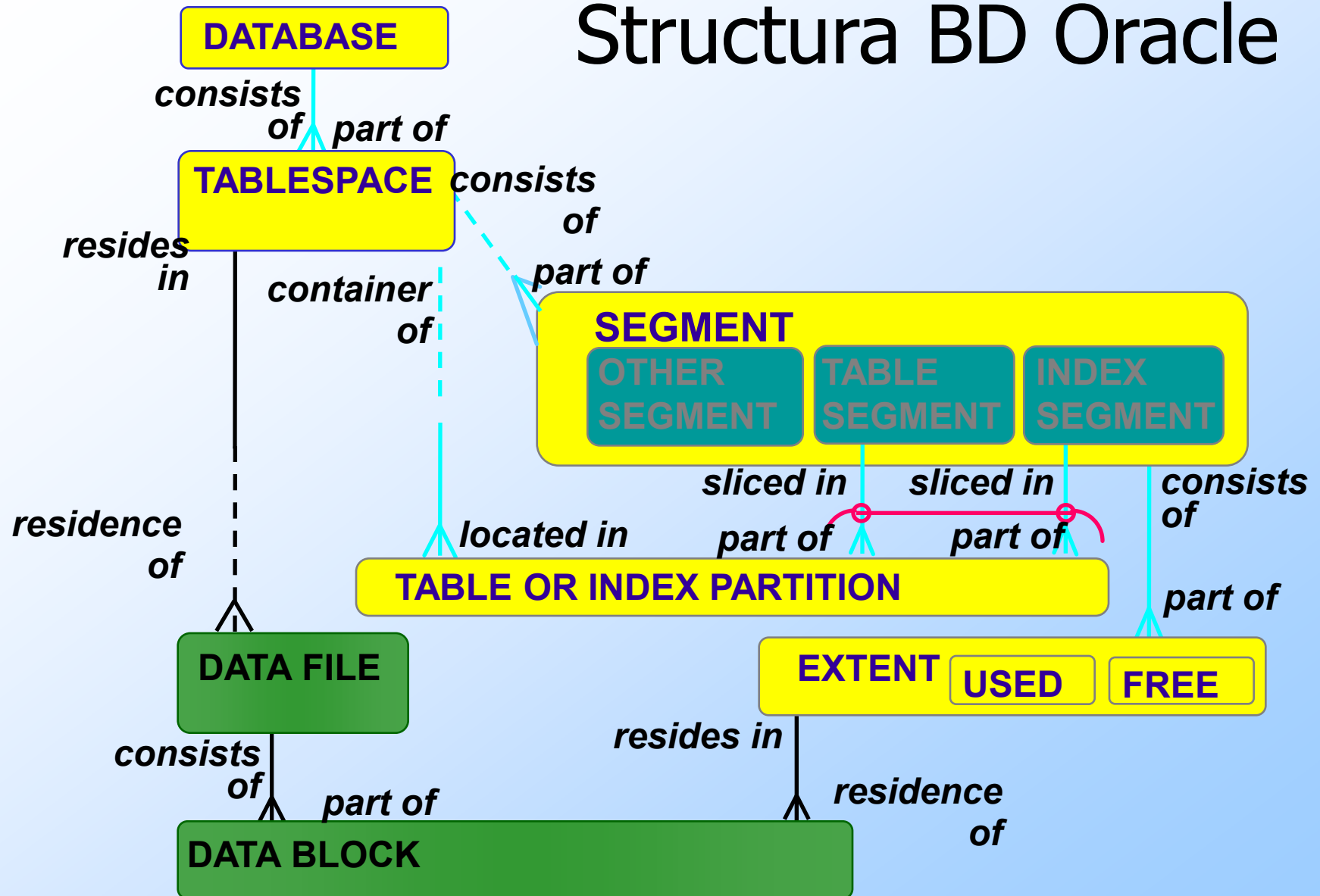
□ Most Commonly-Used Oracle Data Types

- 7. **CLOB** Character large object type. Replaces LONG. Major difference: a table can have more than one CLOB column where there was only one LONG allowed. Maximum size is 4 gigabytes.
 - **Typical use:** see LONG.
- 8. **BLOB** Character large object type. Replaces LONG RAW. Major difference: a table can have more than one BLOB column where there was only one LONGRAW allowed. Maximum size is 4 gigabytes.
 - **Typical use:** see LONG RAW.
- 9. **BFILE** Contains a locator to a large binary file stored outside the database to enable byte stream I/O access to external LOBs residing on the database server.
 - **Typical use:** movies

Column Sequence

- The sequence of columns in a table is relevant, although any column sequence would allow all table operations. The column sequence can influence, in particular, the performance of data manipulation operations. It may also influence the size of a table.
- The suggested optimal column sequence is the following:
 - Primary key columns
 - Unique key columns
 - Foreign key columns
 - Remaining mandatory columns
 - Remaining optional columns
- In cases where the table contains a LONG or LONG RAW column, even if it is a mandatory column, make it the last column of the table.
- The rationale is that null columns should be at the end of the table; columns that are often used in search conditions should be up front. This is for both storage and performance reasons.

Structura BD Oracle



Oracle Database Structure

□ **Tablespaces**

- An Oracle database consists of one or more tablespaces. Each tablespace can hold a number of segments, and each segment must be wholly contained in its tablespaces. The SYSTEM tablespace is created as part of the database creation, and should be reserved for the Oracle Data Dictionary and related tables only. You should not create application data structures in this tablespace. You are advised to create separate tablespaces for different types of segments.

□ **Segments**

- A segment is the space occupied by a database object. There are three types of segments: a table segment, an index segment or an other segment, that is used for clusters. Only the other segments must be part of one tablespace.

□ **Partitions**

- Usually, a segment is assigned to a single tablespace. However, with Oracle8 it is possible to spread a table or index segment into more than one tablespace. This technique is called partitioning. A partition is the part of a table segment (or index segment) that resides in one tablespace.

Oracle Database Structure

□ **Extents**

- Each time more space is needed by a segment, a number of contiguous blocks is allocated as an extent. There is no maximum limit on the number of extents that can be allocated to a segment. It is usually preferable to avoid an excessive number of small extents by ensuring that the segment has a sufficiently large initial extent.

□ **Data Files**

- Data files are the operating system files that physically contain the database data. Data files consist of data blocks.

□ **Data Blocks**

- A data block is the smallest amount of data Oracle reads in one read operation. A data block always contains information from one extent only.
- There is a distinction between the logical table, made up of rows with columns, and the physical table, taking space that is made up of database blocks organized in extents and located in data files.

Database Types

