Oracle Database Design - 2024

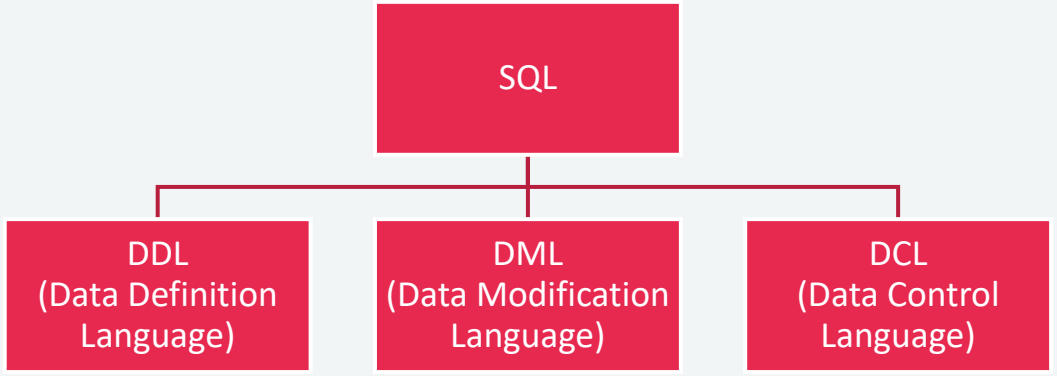
KRYSTIAN WOJTKIEWICZ

SQL – Structured Query Language

- SQL1 – ISO 1987
- SQL2 - ISO 1992
- SQL3 - ISO 1999
- SQL 2003
- SQL 2006
- SQL 2008
- SQL 2011
- SQL 2016
- SQL 2019
- SQL 2023

A commercial implementation of the SQL is called **dialect.**

The environment in which the database is stored and manipulated is called **Database Management System (DBMS)**.



Relational tables

- Relational tables are created, modified, and deleted using DDL commands of SQL language.
- The structure of the command that creates the table reflects the structure of the table specified in logical model of the database.
- Within the command syntax, restrictions on the table attributes and the entire structure of the created database (integrity bonds) are also defined

mandatory attributes

unique attributes

domain bonds

entity integrity

reference integrity

propagation bonds

general bonds

Relational tables

CREATE TABLE Relation_name
({attribute_name attribute_type [{**NOT NULL**} | **NULL**]
[**DEFAULT** default_value]
[{attribute_constraint [...]}] [, ...]}
[, {relation_constraint [, ...]}]);

The basic types of relational table attributes

Type name	Type description
CHAR	Fixed-length character strings (usually the default length is 255 characters)
CHAR(w)	Character strings with fixed length w (1≤w≤2000)
VARCHAR2	Character strings of variable length, however, not more than 4000 characters
VARCHAR2(w)	Character strings of variable length but not more than w characters (w≤4000)
LONG	Character data of variable length (up to 2 GB). Oracle recommends using CLOB
CLOB	Large character data (up to 128 TB, Oracle versions older than 10g to 4 GB)
NUMBER	Integers with a precision of 38 (precision is the number of significant digits)
NUMBER(w)	Integers with maximum precision w (w≤38)

Type name	Type description
NUMBER(w,d)	Real numbers with precision w and d decimal places (w≤38)
DATE	Date and time.
RAW	Binary data up to 2 KB in size
LONG RAW	Binary data up to 2 GB in size. Oracle recommends using BLOB.
ROWID	Represents the specific row address in the table
BLOB	Large volume binary data (up to 4 GB)
BFILE	Binary files. It allows to store read-only binary data in external files (outside the database)
XMLTYPE	Stores XML documents in CLOB columns (since Oracle 9i). Defined by the SYS module (SYS.XMLTYPE)
User data types	User-defined complex types (from Oracle 9i) using the base types specified above and other previously defined complex types

Constraint syntax	Description of constraint
PRIMARY KEY [({attribute_name [, …]})]	For attribute constraint, it defines the attribute that acts as the primary key (entity integrity). For a compound key (relation constraint), a list of key attributes is specified. The constraint excludes with the UNIQUE constraint.
UNIQUE [({attribute_name [, …]})]	For attribute constraint, it defines an attribute that acts as a unique key (with NOT NULL constraint, it will be an alternative key). For a compound key (relation constraint), a list of key attributes is specified. The constraint excludes with the PRIMARY KEY consytaint.
[**FOREIGN KEY** ({attribute_name [, …]})] **REFERENCES** relation_name [({attribute_name [, …]})]	Defines a foreign key (reference integrity). For the attribute constraint (simple foreign key), there is no FOREIGN KEY clause that lists the attributes that make up the key. Relation name means a name of the related relation followed by a list of attributes of this relation that act there as the primary, alternative, or unique key. For a simple key, this is a one-item list.
ON DELETE {**CASCADE** | **SET NULL**}	Constraint following the definition of a foreign key. It defines the binding propagation constraints (CASCADE - cascade deletion, SET NULL - deleting with NULL values inserting). If this constraint no occurs, limited deletion applies.
CHECK (condition)	The constraint concern domain constraints. It defines a condition that must be met by an attribute (attribute constraint) or several attributes (relation constraint) of the relation. In some SQL dialects condition can contain the subquery (Oracle does not do this).

The attribute constraint and relation constraints syntax

[**CONSTRAINT** constraint_name] constraint

Example

Task 1. *Define a relational table* `Persons` *with attributes:* `pesel`, `surname`, `first_name`, `gender` *with appropriate restrictions.*

```
CREATE TABLE Persons
(pesel NUMBER(11) CONSTRAINT pe_pk PRIMARY KEY
                  CONSTRAINT pe_pe_ch CHECK(pesel>10000000000),
 surname VARCHAR2(20) CONSTRAINT pe_sur_nn NOT NULL,
 first_name VARCHAR2(15) CONSTRAINT pe_fn_nn NOT NULL,
 gender CHAR(1) CONSTRAINT pe_ge_ch CHECK(gender IN ('W','M'))
);
```

Task 2. *Define a relational table* `Items_in_dok` *with attributes* `document_no`, `item_no`, `position_content` *with appropriate restrictions*

```
CREATE TABLE Items_in_dok
(document_no VARCHAR2(20),
 item_no NUMBER(5),
 position_content LONG CONSTRAINT pos_con_nn NOT NULL,
 CONSTRAINT pos_con_pk PRIMARY KEY(document_no, item_no)
);
```

Example

Task 3.1. *Let the data of honorary blood donors be collected in the* `Donors` *relation, data of donations collected from them in the* `Donations` *relation, and the results of virological tests of donations in the* `Test_results` *relation. Define the listed relations (attributes, attribute constraints, and relevant relationships between them).*

```
CREATE TABLE Donors
(donor_no NUMBER(5) CONSTRAINT don_pk PRIMARY KEY,
 surname VARCHAR2(20) CONSTRAINT don_sur_nn NOT NULL,
 first_name VARCHAR2(15) CONSTRAINT don_fn_nn NOT NULL,
 gender CHAR(1) CONSTRAINT don_ge_nn NOT NULL
                CONSTRAINT don_pge_ch CHECK(gender IN ('W','M')),
 blood_group CHAR(2) CONSTRAINT don_blg_ch
                     CHECK (blood_group IN ('0','A','B','AB')),
 address VARCHAR2(50) CONSTRAINT don_add_nn NOT NULL
);
```

Example

Task 3.2. *Let the data of honorary blood donors be collected in the* `Donors` *relation, data of donations collected from them in the* `Donations` *relation, and the results of virological tests of donations in the* `Test_results` *relation. Define the listed relations (attributes, attribute constraints, and relevant relationships between them).*

```
CREATE TABLE Donations
(donation_no VARCHAR2(20) CONSTRAINT dona_pk PRIMARY KEY,
 volume NUMBER(3) CONSTRAINT dona_vo_nn NOT NULL
                  CONSTRAINT dona_vo_ch CHECK (volume>0),
 collection_date DATE CONSTRAINT dona_colld_nn NOT NULL,
 donor_no NUMBER(5) CONSTRAINT dona_drno_nn NOT NULL
                    CONSTRAINT dona_drno_fk
                               REFERENCES Donors(donor_no)
);
```

Example

Task 3.3. *Let the data of honorary blood donors be collected in the* `Donors` *relation, data of donations collected from them in the* `Donations` *relations, and the results of virological tests of donations in the* `Test_results` *relation. Define the listed relations (attributes, attribute constraints, and relevant relationships between them).*

```
CREATE TABLE Test_results
(test_id VARCHAR2(15) CONSTRAINT tr_pk PRIMARY KEY,
 result_wr CHAR(1) CONSTRAINT tr_rwr_ch
                   CHECK (result_wr IN ('-', '+','?')),
 result_hiv CHAR(1)  CONSTRAINT tr_rhiv_ch
                     CHECK (result_hiv IN ('-','+','?')),
 result_hbs CHAR(1)  CONSTRAINT tr_rhbs_ch
                     CHECK (result_hbs IN ('-','+','?')),
 result_ahcv CHAR(1) CONSTRAINT tr_rahcv_ch
                     CHECK (result_ahcv IN ('-','+','?')),
 test_data DATE,
 donation_no VARCHAR2(20) CONSTRAINT tr_dono_nn NOT NULL
                          CONSTRAINT tr_dono_fk
                             REFERENCES Donations(donation_no)
                             ON DELETE CASCADE
);
```

Example – alternative

Task 3.2.a *Let the data of honorary blood donors be collected in the* `Donors` *relation, data of donations collected from them in the* `Donations` *relation, and the results of virological tests of donations in the* `Test_results` *relation. Define the listed relations (attributes, attribute constraints, and relevant relationships between them).*

```
CREATE TABLE Donations
(donation_no VARCHAR2(20),
 volume NUMBER(3) CONSTRAINT dona_vo_nn NOT NULL,
 collection_date DATE CONSTRAINT dona_vo_nn NOT NULL,
 donor_no NUMBER(5) CONSTRAINT dona_vo_nn NOT NULL,
 CONSTRAINT dona_pk PRIMARY KEY (donation_no),
 CONSTRAINT dona_vo_ch CHECK (volume>0),
 CONSTRAINT dona_drno_fk FOREIGN KEY (donor_no)
                         REFERENCES Donors(donor_no)
 );
```

Relational tables

ALTER TABLE Relation_name
{**ADD** attribute_name attribute_type
[{**NOT NULL**} | **NULL**]
[**DEFAULT** default_value] attribute_constraint }
| {**DROP COLUMN** attribute_name
[**CASCADE CONSTRAINTS**}
| {**ADD** relation_constraint }
| {**DROP** {constraint_sort |
CONSTRAINT constraint_name}}
| {**DISABLE** | **ENABLE**} { constraint_sort |
CONSTRAINT constraint_name }
| {**MODIFY** attribute_name attribute_type
[**DEFAULT** default_value]
[{**NULL** | **NOT NULL**}]}

Relational tables – ALTER TABLE

- add a new attribute to the relation,
- remove an attribute from the relation. *CASCADE CONSTRAINTS clause removes the attribute from all objects that refer to the removed attribute*,
- add relation constraint,
- remove a constraint by its sort (available sorts are UNIQUE and PRIMARY KEY) or its name,
- temporarily disable or enable the constraint by its sort (available sorts are UNIQUE, PRIMARY KEY, and ALL TRIGGERS) or name,
- modify the attribute specifying its default value and/or mandatory status.

Example

Task 4. *Extend the* `Donor` *relation with a new attribute* `donor_status`*, specifying whether the donor is active (status 'A') or is a retired person (status 'R') or is already dead (status 'D').*

```
ALTER TABLE Donors ADD donor_status CHAR(1) DEFAULT 'A'
                       CONSTRAINT don_sta_ch
                       CHECK (donor_status IN ('A','R','D'));
```

Task 5*. Add a constraint that does not allow entering a donation larger than 999 ml for the* `volume` *attribute in the* `Donations` *relation.*

```
ALTER TABLE Donations
      ADD CONSTRAINT dona_vol_ch CHECK (volume<=999);
```

Example

Task 6. *Remove from the* `Donations` *relation the constraint named* ***dona_vol_ch*** *for the* `volume` *attribute.*

```
ALTER TABLE Donations DROP CONSTRAINT dona_vol_ch;
```

Relational tables

DROP TABLE Relation_name **[CASCADE CONSTRAINT]**

Relational tables

SELECT [DISTINCT | ALL] {expression [alias] [, ...]} | *
FROM {RelationViewName [alias]
[join_operator RelationViewName [alias]
[ON joining_condition]] [, ...]}
[WHERE row_selection_condition**]**
[GROUP BY {expression [, ...]**}**
[HAVING group_selection_condition]**]**
[ORDER BY {expression [DESC | ASC] [, ...]}**]**

Relational tables

Example

Task 7. *List the values of all attributes of the* `Functions` *relation*

```
SELECT * FROM Functions;
```

```
FUNCTION   MIN_MICE             MAX_MICE
---------- -------------------- ------------------
BOSS       90                   110
THUG       70                   90
CATCHING   60                   70
CATCHER    50                   60
CAT        40                   50
NICE       20                   30
DIVISIVE   45                   55
HONORARY   6                    25
```

8 rows selected