

A thick black L-shaped frame surrounds the text. The top-left corner is a horizontal bar extending to the right, then a vertical bar extending downwards. The bottom-right corner is a horizontal bar extending to the left, then a vertical bar extending upwards.

DATABASE DESIGN & IMPLEMENTATION

ICT Skills

Objectives

- Learn about 'constraints' as they relate to data integrity.
- Know when it is possible to define a constraint at the column level and when it is possible at the table level.
- Know what data integrity rules are enforced by NOT NULL and UNIQUE constraints.

Constraints

- A constraint is a database rule.
- All constraint definitions are stored in the data dictionary.
- Constraints prevent the deletion of a table if there are dependencies from other tables.
- Constraints enforce rules on the data whenever a row is inserted, updated, or deleted from a table.
- Constraints are important and so is naming them appropriately.

Constraints

- We can define our constraints in our CREATE TABLE statement.
- There are two place we can define constraints in the CREATE TABLE statement,
 - *At the column level next to the name and the data type*
 - *At the table level after all the column names are listed.*
- The column level refers to where the columns are defined.
- The table level refers to the last line in the statement below the list of columns.

Column Level Constraints

- A column level constraint references a single column.
- It must be defined in the CREATE TABLE statement as part of the column definition.

```
CREATE TABLE clients
(client_number NUMBER(4) CONSTRAINT clients_client_num_pk PRIMARY KEY,
first_name      VARCHAR2(14),
last_name       VARCHAR2(13));
```

- The name of the constraint is clients_client_num_pk.
- It enforces the business rule that the client_number is the primary key of the clients table.

Naming Constraints

- Every constraint in the database has a name. When a constraint is created it does not have to be given a name by the author of the CREATE TABLE statement, in which case the system gives the constraint a name such as SYS_C0058534.
- A naming convention can be the combination of the table name, a column name and the type of constraint.
 - *Pk for primary key, nn for not null, uk for unique key, fk for foreign key.*
- If the reserved word CONSTRAINT is used in the CREATE TABLE definition, you must give the constraint a name (max 30 chars).
- It is best to name constraints as system level names are not easy to interpret.

Naming Constraints

- Here is an example:

```
CREATE TABLE clients
(client_number NUMBER(4) CONSTRAINT clients_client_num_pk PRIMARY KEY,
last_name      VARCHAR2(13) NOT NULL,
email         VARCHAR2(80));
```

Table Level Constraints

- Table level constraints are listed separately from the column definitions in the CREATE TABLE statement after all columns are defined.

```
CREATE TABLE clients (  
  client_number NUMBER(6) NOT NULL,  
  first_name     VARCHAR2(20),  
  last_name      VARCHAR2(20),  
  phone          VARCHAR2(20),  
  email          VARCHAR2(10) NOT NULL,  
  CONSTRAINT clients_phone_email_uk UNIQUE (email,phone));
```

- You must include the column(s) name for which the constraint is being defined.

Rules for Constraints

- Constraints that refer to more than one column must be defined at the table level.
- The NOT NULL constraint can be defined only at the column level.
- UNIQUE, PRIMARY, FOREIGN KEY, and CHECK constraints can be defined at either level.
- If the keyword CONSTRAINT is used in the CREATE TABLE statement, you must provide a name for the constraint.

Constraints Violations

```
CREATE TABLE clients(  
    client_number    NUMBER(6),  
    first_name       VARCHAR2(20),  
    last_name        VARCHAR2(20),  
    phone            VARCHAR2(20) CONSTRAINT phone_email_uk  
                    UNIQUE(email,phone),  
    email            VARCHAR2(10) CONSTRAINT NOT NULL,  
    CONSTRAINT emailclients_email NOT NULL,  
    CONSTRAINT clients_client_num_pk PRIMARY KEY (client_number));
```

Constraint Types

- Five types of constraints:
 - *NOT NULL*
 - *UNIQUE*
 - *PRIMARY KEY*
 - *FOREIGN KEY*
 - *CHECK*

NOT NULL & UNIQUE Constraints

- A column defined with a NOT NULL constraint requires that for every row entered into the table, a value must exist for that column.
- It is customary to name a NOT NULL constraint using `_nn`
- A UNIQUE constraint requires that every value in a column or set of columns (composite) be unique; no two rows can have the same values.
- It is customary to name a UNIQUE constraint using `_uk`
- To define a composite UNIQUE constraint you must do it at the table level
- UNIQUE key constraint allows NULL's unless the column also has a NOT NULL constraint.

Constraints

CLIENT_NUMBER	FIRST_NAME	LAST_NAME	PHONE	EMAIL
5922	Hiram	Peters	3715832249	hpeters@yahoo.com
5857	Serena	Jones	7035335900	serena.jones@jones.com
6133	Lauren	Vigil	4072220090	lbv@lbv.net

```
INSERT INTO clients (client_number, first_name, Last_name, phone,  
email)  
VALUES (7234, 'Lonny', 'Vigil', 4072220091, 'lbv@lbv.net');
```

```
ORA-00001: unique constraint  
(USWA_SKHS_SQL01_T01.CLIENT_EMAIL_UK) violated
```

Constraints

CLIENT_NUMBER	FIRST_NAME	LAST_NAME	PHONE	EMAIL
5922	Hiram	Peters	3715832249	hpeters@yahoo.com
5857	Serena	Jones	7035335900	serena.jones@jones.com
6133	Lauren	Vigil	4072220090	lbv@lbv.net
7234	Lonny	Vigil	4072220091	lbv@lbv.net



This combination of columns
must be **UNIQUE**

```
CONSTRAINT clients_phone_email_uk UNIQUE(email,phone)
```