Data base:

Data types:

* smallint : 2 bytes integer
* integer : 4 bytes - integer
* real : Real number
* double precision:

Decimal types:

Decimal

Numeric.

* char (Length) . With the same
* Varchar (Max length)

Date types:

* Date .
* time : hour, minutes, seconds.
* timestamp - precised moment.
* interval : time interval

Binary types:

Bit: save one bit.

Bit(5) / Bit(3) - etc.

Langage définition de donnée : (LDD)

Create an entity :

create table table_name (colonne type ...);

default keyword, initialize the attribute with the given data.

Example :

```
create table thing (
    name varchar(100) default 'Alan');
```

- to get the schema of the entity:

describe entity's name;

constraints in sql:

. possible constraints:

primary key, Foreign key,

References, Unique, check ...

. on the entity:

create table, Alter table...

→ you can give the constraints

names.

example of giving a constraint while creating the entity:

create tablespace(id integer constrain [constraint name] primary key(Attn1, 2 ...)

- External key:

```
create table emp (
. . .,
dept integer constraint r_dept references
dept(dept))
```

optional if the target is primary key.

should add

```
create table emp (
. . .,
dept integer,
constraint r_dept foreign key(dept)
references dept(dept))
```

Foreign key.

Check : you know that :

- for the constraints, it is possible

to enable / Disable constraints.

```
ALTER TABLE emp
DROP CONSTRAINT nom_unique
ADD CONSTRAINT sal_min CHECK(sal + coalesce(comm,0)>50)
RENAME CONSTRAINT nom1 TO nom2
MODIFY CONSTRAINT sal_min DISABLE
```

to watch on youtube :

- constraints,
- controlling them,
- verifying
- update

Learn crud shits:

basics : insert, update, delete.

---

dealing with null values.

sql command coalesce(_, _, _)

↑

values.

Returns the first non nullable element.

```
SELECT COALESCE(column1, column2, 'DefaultValue') AS
Result
FROM your_table;
```

- Command nullif:

Returns if the 2 elements in the arguments are equal ; otherwise the value of the first element.

```
SELECT NULLIF(column1, 0) AS Result
FROM your_table;
```

Sql Joins.

1) Natural Join :

⇒ combines rows from two tables based on common column names.

- How it works:
  - identify columns with similar names.
  - Eliminates duplicated data. of columns.
  - combine the Rows.

Example 1 :

table (1) : employee

EmployeeID	Name	DepartmentID
1	Alice	101
2	Bob	102
3	Charlie	101

table 2 : departement

DepartmentID	DepartmentName
101	HR
102	IT
103	Finance

Command sql :

```
SELECT *
FROM Employees
NATURAL JOIN Departments;
```

Result :

EmployeeID	Name	DepartmentID	DepartmentName
1	Alice	101	HR
2	Bob	102	IT
3	Charlie	101	HR