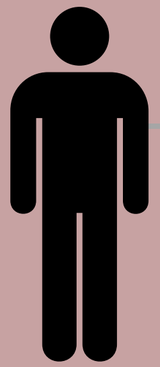
Database - SQL

Septembre 2022

Rappels

Rappels



Nom
Prénom
Adresse
Date de naissance
...

EJOUR 2

SEJOUR 3

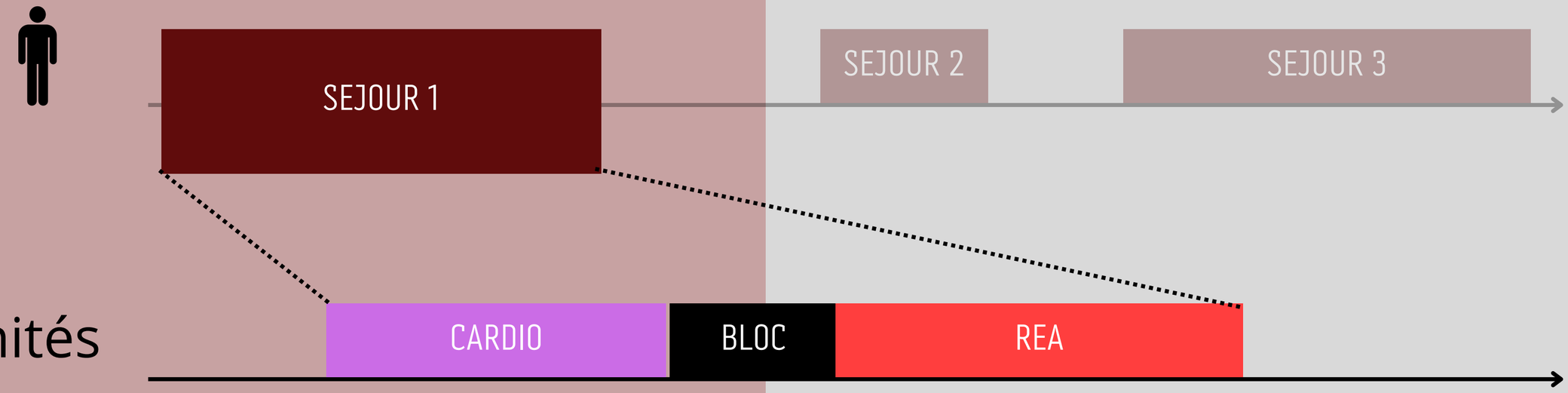
Rappels

SEJOUR 1

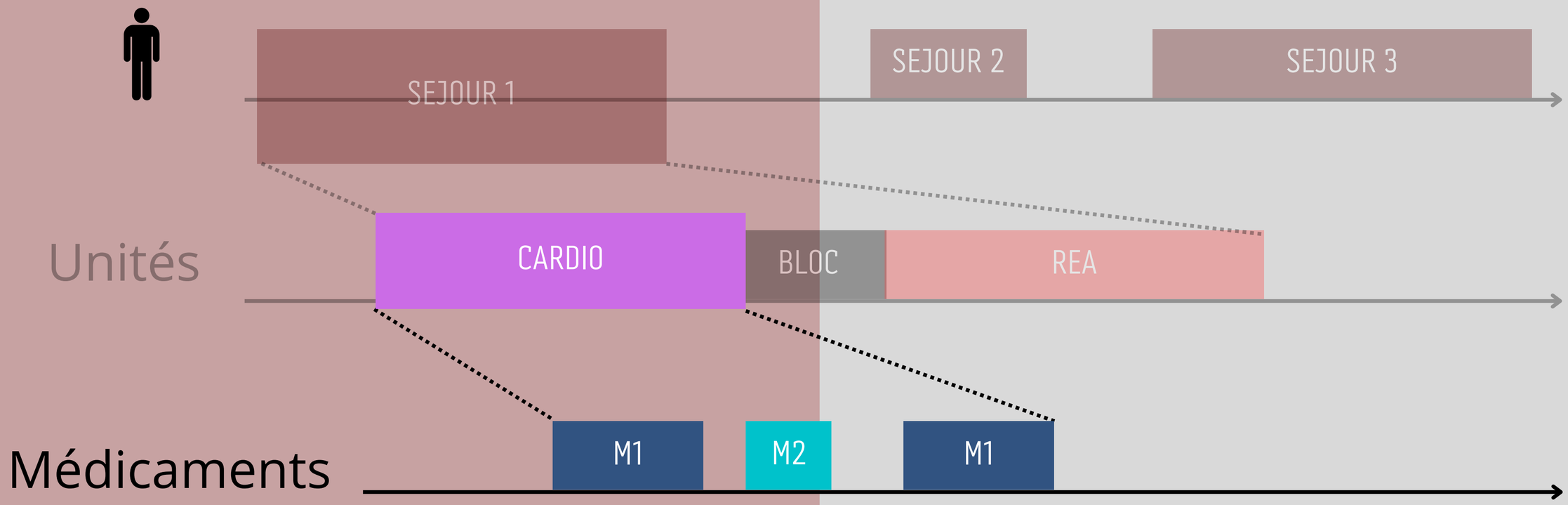
Date d’entrée
Date de sortie

SEJOUR 3

Rappels



Rappels


SEJOUR 1
SEJOUR 2
SEJOUR 3
Unités
CARDIO
BLOC
REA
M1
M2
M1
Médicaments

Rappels

SQL

Langage permettant d'exploiter une base de données

DBMS

Système de gestion des base de données

Relations

Contraintes (clés primaires, clés secondaires)

Rappels

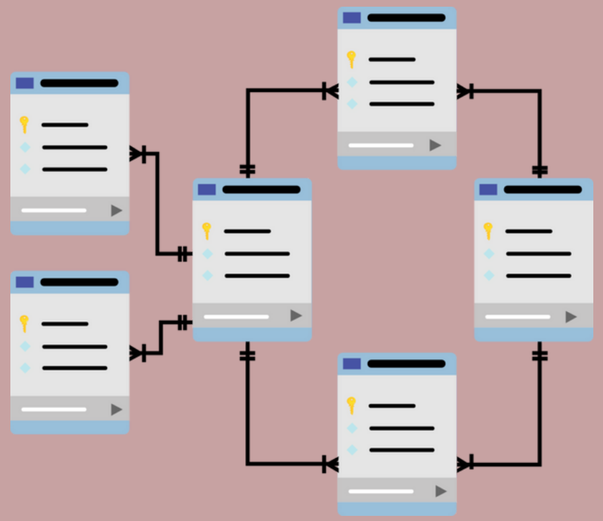
Relations

Contraintes (clés primaires, clés secondaires)

PATIENT_ID	NAME	SEXE
1	Virgil	M
2	Carmela	F
3	Bruno	M

STAY_ID	PATIENT_ID	UNIT
1	1	U3
2	1	U3
3	3	U4

Rappels



SQL

Langage permettant d'exploiter une base de données

DBMS

Système de gestion des base de données

PostgreSQL

MariaDB

Relations

Contraintes (clé primaire, clé secondaires)

Modeling

Lier les clés entre les différentes tables

Vocabulaire

Table de vocabulaire (données en texte lié à un ID)

Rappels

Vocabulaire

Table de vocabulaire (données en texte lié à un ID)

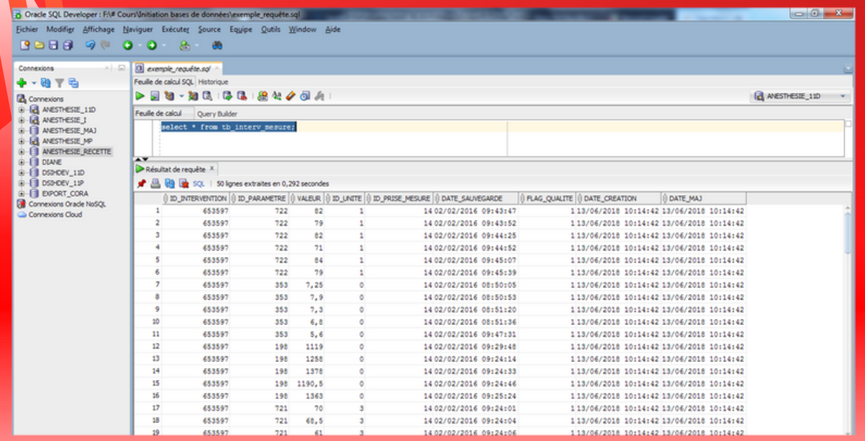
PATIENT_ID	DRUG_ID	DATE
1	**26**	01/01/2022
2	**123**	01/01/2022
3	**123**	02/01/2022

DRUG_ID	DRUG_LABEL
26	paracetamol
123	insulin
13	aspirin

REQUÊTES SQL

```
MariaDB [cours]> select * from patient;
ERROR 1146 (42S02): Table 'cours.patient' doesn't exist
MariaDB [cours]> SELECT * FROM PATIENT;
+------------+--------------+-----------+------------+-----+
| PATIENT_ID | FIRST_NAME   | LAST_NAME | BIRTH_DATE | SEX |
+------------+--------------+-----------+------------+-----+
|          1 | First name 1 | Name 1    | 2000-03-30 | F   |
|          2 | First name 2 | Name 2    | 1980-02-20 | F   |
|          3 | First name 3 | Name 3    | 1972-03-05 | H   |
|          4 | First name 4 | Name 4    | 1938-09-20 | H   |
|          5 | First name 5 | Name 5    | 1940-10-01 | F   |
|          6 | First name 6 | Name 6    | 1965-05-06 | F   |
|          7 | First name 7 | Name 7    | 1967-06-07 | H   |
|          8 | First name 8 | Name 8    | 1981-03-31 | F   |
|          9 | First name 9 | Name 9    | 2010-03-12 | F   |
|         10 | First name 10 | Name 10  | 2020-03-10 | H   |
|         11 | First name 11 | Name 11  | 1981-02-20 | F   |
|         12 | First name 12 | Name 12  | 1973-03-05 | H   |
```

Terminal



Editor (DBeaver)

```php
<?php

$reponse = $bdd->query('SELECT p.IEP, j.* FROM jour j, patient p WHERE p.ID_INCLUS
                        and j.ID_INCLUSION = "'.$ID_INCLUSION.'"');
    if(!empty($reponse))
        {
            ?>
            <table>
                <tbody>

            <?php

            while ($donnees = $reponse->fetch())
            {

            ?>

                <tr>
                    <td>Jour <?php echo $donnees['ID_JOUR']; ?></td>
```

Application

REQUÊTES SQL

DDL (Data Definition Language)

Modifier la structure/architecture des tables de la base de données

DML (Data Manipulation Language)

Modifier ou manipuler les enregistrements dans une table

DCL (Data Control Language)

Contrôle l'accès aux données

SQL commandes

DDL

- CREATE
- DROP
- ALTER
- TRUNCATE
- RENAME
- TABLE
- DATABASE

DML

- SELECT
- INSERT
- DELETE
- UPDATE
- WHERE
- GROUP BY
- ORDER BY

DCL

- GRANT
- DENY
- REVOKE
- COMMIT
- ROLLBACK

Pratique - DDL

```
CREATE TABLE PERSON (
--
PERSON_ID      INT           NOT NULL AUTO_INCREMENT,
FIRST_NAME     VARCHAR(50)   NOT NULL,
LAST_NAME      VARCHAR(50)   NOT NULL,
BIRTH_DATE     DATE          NOT NULL,
SEX            CHAR(1)       NOT NULL,
--
PRIMARY KEY (PERSON_ID)
)
COLLATE='utf8_general_ci'
AUTO_INCREMENT=1
;
```

Pratique - DDL

```
CREATE TABLE VISIT_OCCURRENCE (
--
VISIT_OCCURRENCE_ID    INT     NOT NULL AUTO_INCREMENT,
PERSON_ID              INT     NOT NULL,
VISIT_START_DATE       DATE    NOT NULL,
VISIT_END_DATE         DATE    NOT NULL,
--
PRIMARY KEY (VISIT_OCCURRENCE_ID),
FOREIGN KEY (PERSON_ID) REFERENCES PERSON (PERSON_ID)
)
COLLATE='utf8_general_ci'
AUTO_INCREMENT=1
;
```

Pratique - DDL

```
CREATE TABLE VISIT_OCCURRENCE (
--
VISIT_OCCURRENCE_ID    INT     NOT NULL AUTO_INCREMENT,
PERSON_ID              INT     NOT NULL,
VISIT_START_DATE       DATE    NOT NULL,
VISIT_END_DATE         DATE    NOT NULL,
--
PRIMARY KEY (VISIT_OCCURRENCE_ID),
FOREIGN KEY (PERSON_ID) REFERENCES PERSON (PERSON_ID)
)
COLLATE='utf8_general_ci'
AUTO_INCREMENT=1
;
```

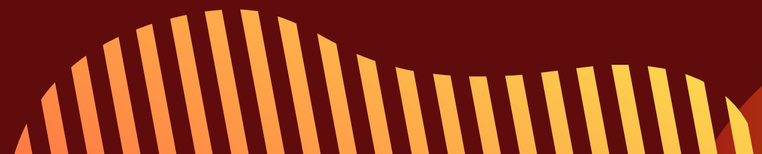
La table PERSON doit déjà avoir été créée

Pratique - DDL

```
ALTER TABLE PERSON
ADD COLUMN IF NOT EXISTS (PERSON_SOURCE_VALUE char(10) default('0')) ;

ALTER TABLE PERSON
MODIFY PERSON_SOURCE_VALUE varchar(10) ;

ALTER TABLE PERSON
DROP COLUMN PERSON_SOURCE_VALUE ;
```



REQUÊTES DML

INSERT

```
INSERT INTO PERSON (`FIRST_NAME`, `LAST_NAME`, `BIRTH_DATE`, `SEX`)
VALUES ('First name 1', 'Name 1', '2000-03-30', 'F') ;
```

```
UPDATE PERSON
SET BIRTH_DATE = '1980-03-24'
WHERE PATIENT_ID = '1';
```

```
DELETE FROM PERSON
WHERE PERSON_ID = '2' ;
```

REQUÊTES DML

SELECT

```
SELECT expr [, expr ...]
FROM table_name
WHERE condition
GROUP BY colonne_name
HAVING condition agrégation
{ UNION | INTERSECT | EXCEPT }
ORDER BY colonne_name
LIMIT count
;
```

REQUÊTES DML

SELECT

```
-- Sélection de tous les champs
SELECT *
FROM PERSON ;
```

```
-- Sélection d'une colonne
SELECT FIRST_NAME
FROM PERSON ;
```

```
-- Alias sur des colonnes
SELECT FIRST_NAME as NAME
FROM PERSON ;
```

```
-- Avec condition
SELECT *
FROM PERSON
WHERE PERSON_ID = 1 ;
```

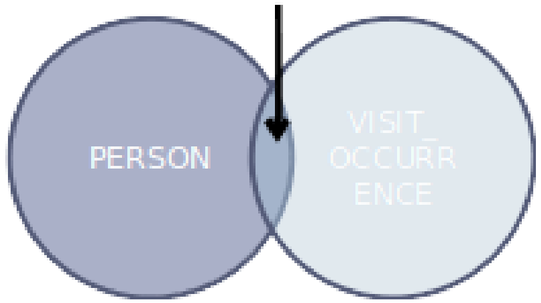
```
-- Avec conditions
SELECT *
FROM PERSON
WHERE PERSON_ID IN (1, 10) ;
```

```
-- Ordonner
SELECT *
FROM PERSON
ORDER BY BIRTH_DATE ;
```

REQUÊTES DML

JOIN

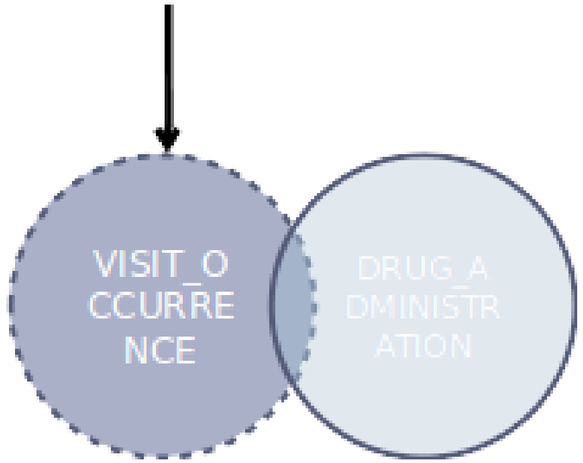
```
SELECT P.PERSON_ID,
        VO.VISIT_START_DATE,
        VO.VISIT_END_DATE
FROM PERSON P
INNER JOIN VISIT_OCCURRENCE VO
WHERE P.PERSON_ID = VO.PERSON_ID
ORDER BY P.PERSON_ID, VO.VISIT_END_DATE ;
```



REQUÊTES DML

JOIN

```
SELECT VO.*
FROM VISIT_OCCURRENCE VO
LEFT OUTER JOIN DRUG_ADMINISTRATION DA
ON VO.VISIT_OCCURRENCE_ID = DA.VISIT_OCCURRENCE_ID ;
```



REQUÊTES DML

Table PERSON

PERSON_ID	SEX
1	F
2	F
3	M
...	...

```
SELECT SEX,
       COUNT(*) AS NB
FROM PERSON
GROUP BY SEX
ORDER BY SEX ;
```

Résultat affiché

SEX	NB
F	11
M	9

REQUÊTES DML

Table PERSON

PERSON_ID	SEX
1	F
2	F
3	M
...	...

```
SELECT SEX,
       COUNT(*) AS NB
FROM PERSON
GROUP BY SEX
ORDER BY SEX ;
```

GROUP BY :

nom de la colonne de regroupement

COUNT(*) :

agrégat des lignes pour le regroupement

```
SELECT SEX,
       COUNT(*) AS NB
       PATIENT_ID
```

Merci

Email

mathilde.frchrt@gmail.com
mathilde.fruchart@univ-lille.fr

GIT

@mathilde.frchrt