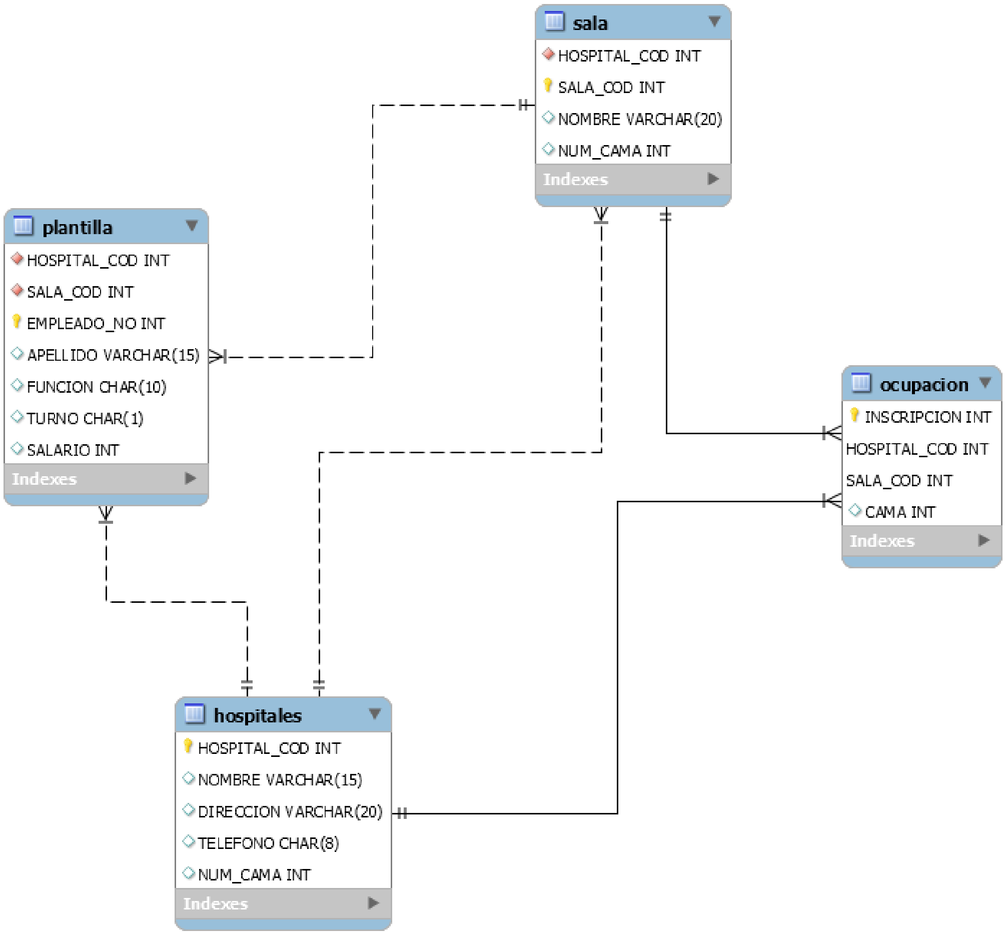
REQUIREMENTS

1. Model A & B **(THIS IS AN EXAMPLE. ONLY ONE MODEL PROVIDED)**
2. Number of queries: 10 queries (basic version 0,75 points + advanced version 1,25 points)
3. Duration of the exam: 2 hours (12 min per query on average)
4. Mistakes on SELECT -0.10 points; Other mistakes -0.25 points
5. Minimum to pass: 4 out of 10 to pass the exam
6. Only one DB; DDL + DML provided; No tips; No detailed output
7. No Internet connection
8. Only workbench, offline docs, one printed/handwritten A4 double sided allowed
9. Handing one single script (plain text) with all queries and optional comments
10. USE DQL DECALOGUE!

MODEL A: HOSPITALS

Find DDL at the end of this document. THIS IS AN EXAMPLE. ONLY DDL PROVIDED.



QUERIES 1A [2 POINTS] CARTESIAN VS JOIN AND FORMATTING THE OUTPUT

1A.1) Requirements (0,75 POINTS):

Using the appropriate aliases, we want to know the maximum wage that exists in each hospital. Show the name of the hospital and the maximum salary.

Usando los alias adecuados, queremos conocer el máximo salario que existe en cada hospital. Muestra el nombre del hospital y su salario medio.

USE CARTESIAN

DO NOT USE "LIMIT"

```
Use a single script
```

1A.2) Requirements (1,25 POINTS):

Using the appropriate aliases, we want to know the maximum wage that exists in each room of each hospital. Show the name of the hospital, the name of the room and the maximum salary.

Usando los alias adecuados, queremos conocer el máximo salario que existe en cada sala de cada hospital.Muestra el nombre del hospital, el nombre de la sala y su salario máximo.

USE JOIN

DO NOT USE "LIMIT"

ADD "EUROS" AFTER THE SALARY AMOUNT

```
Use a single script
```

QUERIES 2A [2 POINTS] SUBQUERIES AND SORTING

2A.1) Requirements (0,75 POINTS):

Using the appropriate aliases, we want to retrieve the number (employee_no) and surname of people who are paid more than the average for their hospital.

Usando los alias adecuados, queremos rcuperar el número (empleado_no) y apellido de las personas que perciban un salario mayor que la media de su hospital.

DO NOT USE "[NOT] EXISTS"

DO NOT USE "[NOT] IN"

```
Use a single script
```

2A.2) Requirements (1,25 POINTS):

Using the appropriate aliases, we want to retrieve the number (employee_no), surname and salary of people who are less paid at each hospital. Place worst salaries first.

Usando los alias adecuados, queremos rcuperar el número (empleado_no), apellido y salario de las personas que perciban el salario más bajo de su hospital. Muestra primero los peor pagados.

USE "[NOT] EXISTS" OR "[NOT] IN"

```
Use a single script
```

THIS IS AN EXAMPLE. ONLY TWO SETS OF QUERIES PROVIDED.

FULL SCRIPT (DDL)

```
DROP DATABASE IF EXISTS HOSPITALES;
CREATE DATABASE HOSPITALES;
USE HOSPITALES;

CREATE TABLE HOSPITAL(
HOSPITAL_COD INTEGER,
NOMBRE VARCHAR(15),
DIRECCION VARCHAR(20),
TELEFONO CHAR(8),
NUM_CAMA INTEGER,
CONSTRAINT HOSPITAL_PK PRIMARY KEY (HOSPITAL_COD)
);

CREATE TABLE SALA(
HOSPITAL_COD INTEGER NOT NULL,
SALA_COD INTEGER,
NOMBRE VARCHAR(20),
NUM_CAMA INTEGER,
CONSTRAINT SALA_PK PRIMARY KEY (SALA_COD),
CONSTRAINT HOSPITAL_SALA_FK FOREIGN KEY (HOSPITAL_COD) REFERENCES HOSPITAL (HOSPITAL_COD)
);

CREATE TABLE PLANTILLA(
HOSPITAL_COD INTEGER NOT NULL,
SALA_COD INTEGER NOT NULL,
EMPLEADO_NO INTEGER,
APELLIDO VARCHAR(15),
FUNCION CHAR(10),
TURNO CHAR(1),
SALARIO INTEGER,
CONSTRAINT EMPLEADO_PK PRIMARY KEY (EMPLEADO_NO),
CONSTRAINT HOSPITAL_FK FOREIGN KEY (HOSPITAL_COD) REFERENCES HOSPITAL (HOSPITAL_COD),
CONSTRAINT SALA_FK FOREIGN KEY (SALA_COD) REFERENCES SALA (SALA_COD),
CONSTRAINT TURNO_CH CHECK (TURNO IN ('T','M','N')),
CONSTRAINT SALARIO_CH CHECK (SALARIO > 0)
);

CREATE TABLE ENFERMO(
INSCRIPCION INTEGER,
APELLIDO VARCHAR(15),
DIRECCION VARCHAR(20),
FECHA_NAC DATE,
S VARCHAR(1),
NSS INTEGER,
CONSTRAINT ENFERMO_PK PRIMARY KEY (INSCRIPCION)
);

CREATE TABLE OCUPACION(
INSCRIPCION INTEGER NOT NULL,
HOSPITAL_COD INTEGER NOT NULL,
SALA_COD INTEGER NOT NULL,
CAMA INTEGER,
CONSTRAINT OC_PK PRIMARY KEY (INSCRIPCION,HOSPITAL_COD, SALA_COD),
CONSTRAINT HOSPITAL_OCUP_FK FOREIGN KEY (HOSPITAL_COD) REFERENCES HOSPITAL
(HOSPITAL_COD),
CONSTRAINT SALA_OCUP_FK FOREIGN KEY (SALA_COD) REFERENCES SALA (SALA_COD)
);
```