# SQL Exercise Sheet 7 bis

1 - Suppose you are given the following requirements for a simple database:

- **A company's product has a number and a name, and reviews are given for this product. Each review has a rating and a comment.**

Construct a conceptual ERD (min-max) and a physical ERD (crow's foot) for this simple database. Additionally, create the tables as it should be done in SQL.

Each review is attributed to 1 and only 1 product, and each product has 1 or many reviews
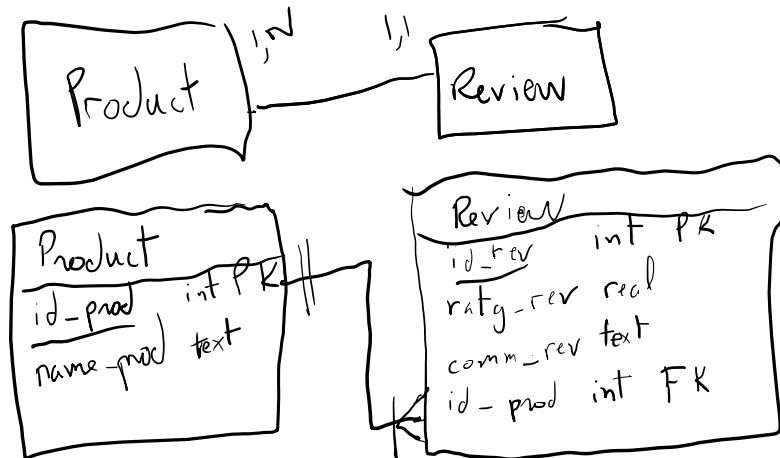

```
CREATE TABLE Product (
    id_prod INT PRIMARY KEY,
    name_prod VARCHAR(100),
);

CREATE TABLE Review (
    id_rev INT PRIMARY KEY,
    ratg_rev FLOAT,
    comm_rev TEXT,
    id_prod INT,
    FOREIGN KEY (id_ prod) REFERENCES Product(id_ prod)
);
```

2 - Suppose you are given the following requirements for a simple database:

- **An employee has a name, an address,**
- **each employee has projects to complete,**
- **each project has a name, and a description.**

Construct a conceptual ERD (min-max) and a physical ERD (crow's foot) for this simple database. Additionally, create the tables as it should be done in SQL.

Each employee may have multiple projects to complete and each project is conducted by 1 or many employees
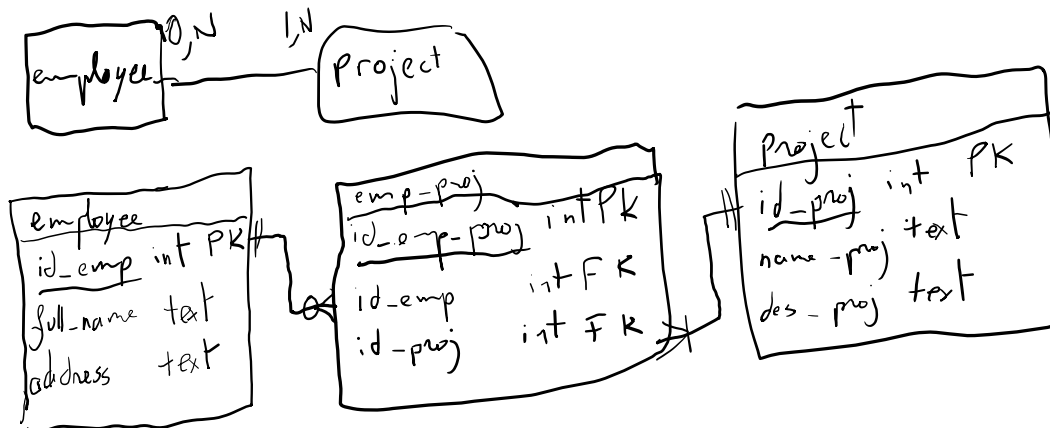

```
CREATE TABLE Employee (
   id_emp INT PRIMARY KEY,
   full_name VARCHAR(100),
   address VARCHAR(500),
);

CREATE TABLE Project (
   id_proj INT PRIMARY KEY,
   name_proj VARCHAR(100),
   des_proj TEXT,
);

CREATE TABLE Emp_proj (
   id_emp_proj INT PRIMARY KEY,
   id_emp INT,
   id_proj INT,
   FOREIGN KEY (id_emp) REFERENCES employee(id_emp)
   FOREIGN KEY (id_ proj) REFERENCES project(id_ proj)
);
```

3 - Suppose you are given the following requirements for a simple database:

- **A person has a first name, a last name and a birth date, and may have a passport that has a number (possibly non-unique) and an expiry date.**

Construct a conceptual ERD (min-max) and a physical ERD (crow's foot) for this simple database. Additionally, create the tables as it should be done in SQL.

Each person can have 0 or 1 passport and each passport is delivered to 1 and only 1 person
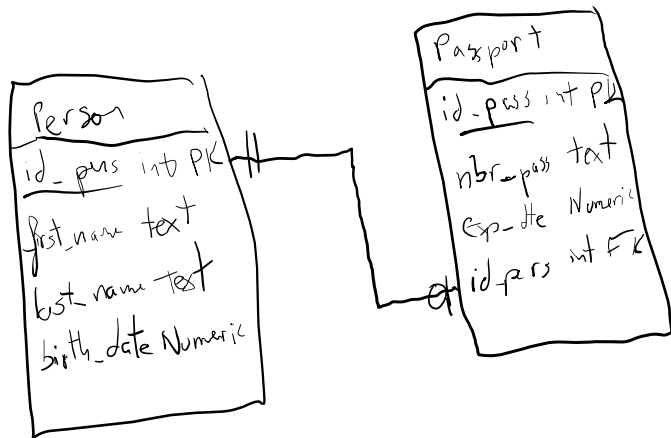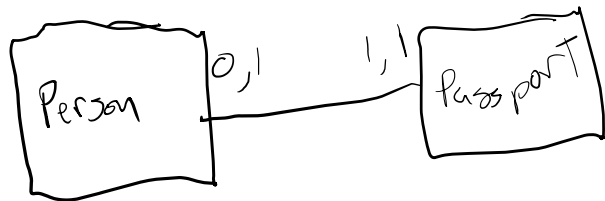

```
CREATE TABLE Person (
   id_pers INT PRIMARY KEY,
   first_name VARCHAR(50),
   last_name VARCHAR(50),
   birth_date DATE,
);

CREATE TABLE Passport (
   id_pass INT PRIMARY KEY,
   nbr_pass VARCHAR(20),
   expiry_date DATE,
   id_pers INT,
   FOREIGN KEY (id_pers) REFERENCES Person(id_pers)
);
```

4 - Suppose you are given the following requirements for a simple database that you have to build for a library:

- A book has a title, a publication year, a number of copies available.
- Each book has a genre, each genre has a name and a description,
- An author has a name, a date of birth, and a nationality,
- A customer has a name and an email address,
- A customer can check out books and a book can be checked out by multiple customers.

Construct a conceptual ERD (min-max) and a physical ERD (crow's foot) for this simple database. Additionally, create the tables as it should be done in SQL.

One-to-One Relationship:
One book has one genre.
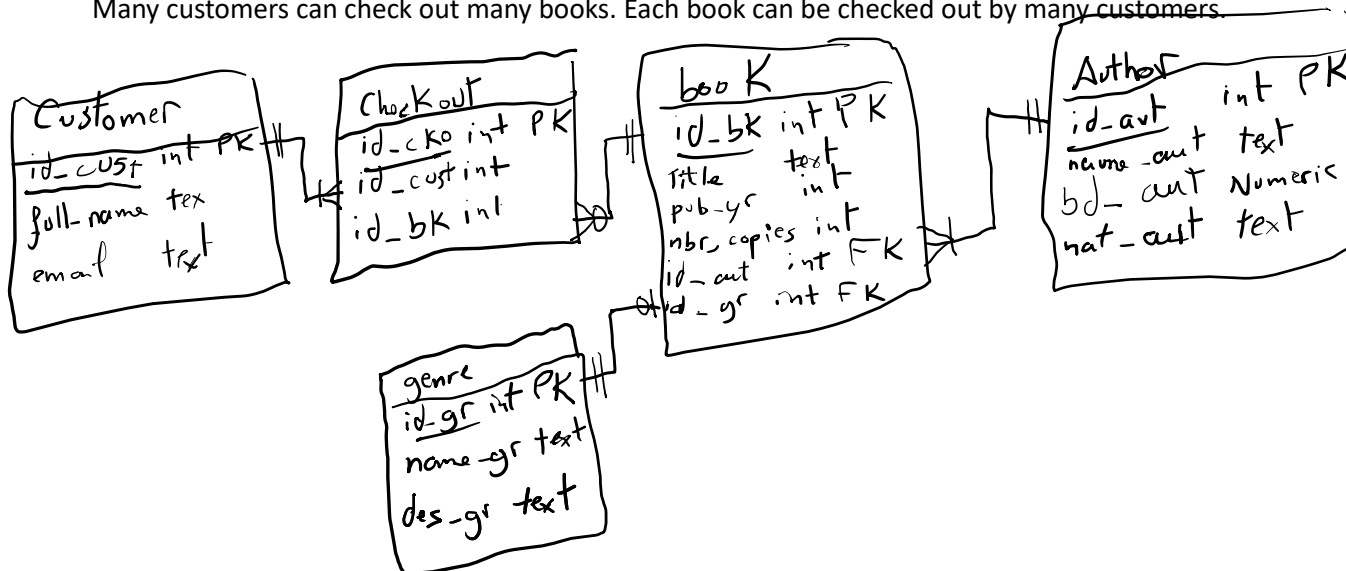One genre is associated with one book.
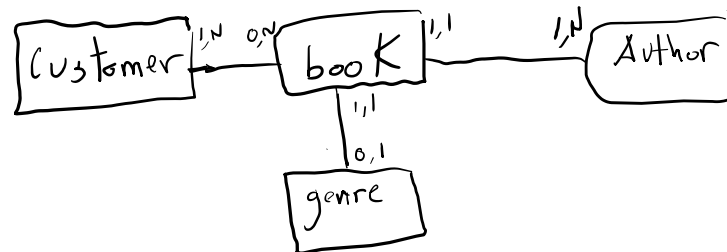
One-to-Many Relationship:
One author can write many books.
Each book is written by one author. (In this example)

Many-to-Many Relationship:
Many customers can check out many books. Each book can be checked out by many customers.

```sql
CREATE TABLE Author (
    id_aut INT PRIMARY KEY,
    name_aut VARCHAR(100),
    Bd_aut DATE,
    nat_aut VARCHAR(50)
);

CREATE TABLE Genre (
    id_gr INT PRIMARY KEY,
    name_gr VARCHAR(50),
    des_gr TEXT
);

CREATE TABLE Book (
    id_bk INT PRIMARY KEY,
    title VARCHAR(100),
    pub_yr INT,
    nbr_copies INT,
    id_aut INT,
    id_gr INT,
    FOREIGN KEY (id_aut) REFERENCES Author(id_aut),
    FOREIGN KEY (id_gr) REFERENCES Genre(id_gr)
);

CREATE TABLE Customer (
    id_cust INT PRIMARY KEY,
    full_name VARCHAR(50),
    email VARCHAR(100)
);
```

```sql
CREATE TABLE Checkout (
    id_cko INT PRIMARY KEY,
    id_cust INT,
    id_bk INT,
    FOREIGN KEY (id_cust) REFERENCES Customer(id_cust),
    FOREIGN KEY (id_bk) REFERENCES Book(id_bk)
);
```