1. **ER Diagram (40 pts):** Notown Records has decided to store information about musicians who perform on its albums (as well as other company data) in a database.

   - Each musician that records at Notown has an SSN, a name, an address, and a phone number. Poorly paid musicians often share the same address, and no address has more than one phone.
   - Each instrument used in songs recorded at Notown has a unique identification number, a name (e.g., guitar, synthesizer, flute) and a musical key (e.g., C, B-flat, E-flat).
   - Each album recorded on the Notown label has a unique identification number, a title, a copyright date, a format (e.g., CD or MC), and an album identifier.
   - Each song recorded at Notown has a title and an author.
   - Each musician may play several instruments, and a given instrument may be played by several musicians.
   - Each album has a number of songs on it, but no song may appear on more than one album.
   - Each song is performed by one or more musicians, and a musician may perform a number of songs.
   - Each album has exactly one musician who acts as its producer. A musician may produce several albums, of course.

Design a conceptual schema for Notown and draw an ER diagram for your schema. The preceding information describes the situation that the Notown database must model. Be sure to indicate all key and cardinality constraints and any assumptions you make. Identify any constraints you are unable to capture in the ER diagram and briefly explain why you could not express them.

**Solution:** WE Diagram is given below

2. **SQL (30 pts):** Write SQL statements for the following queries using the following database and verify your queries using sqlplus.

   Classes(cclass, type, country, numguns, bore, displacement)
   Ships(sname, sclass, launched)
   Battles(sname, date)
   Outcomes(ship, battle, result)

Ships are built in classes from the same design, and the class is usually named for the first ship of that class. The relation Classes records the name of the class, the type (bb for battleship
and bc for battlecruiser), the country that built the ship, the number of main guns, the bore (diameter of the gun barrel, in inches) of the main guns, and the displacement (weight in tons). Relation Ships records the name of the ship, the name of oys class, and the year in which the ship was launched. Relation Battles gives the name and date of battles involving these ships, and relation Outcomes gives the result (sunk, damaged, or ok) for each ship in each battle.

(a) Find the ships heavier than 35,000 tons.

```sql
select sname
from classes, ships
where cclass = sclass
    and displacement > 35000
```

(b) List the name, displacement, and the number of guns of the ships engaged in the battle.
of Guadalcanal.

```sql
select sname, displacement, numguns
from classes, ships, outcomes
where sname = ship
    and sclass = cclass
    and battle = 'Guadalcanal'
```

(c) Find the ships that were damaged in one battle, but later fought in another.

```sql
select ship
from outcomes o1, battles b1
where result = 'damaged'
    and o1.battle = b1.sname
    and exists (select *
from outcomes o2, battles b2
where o2.battle = b2.sname
    and b1.bdate < b2.bdate)
```

(d) Find the countries who has the heaviest ships.

```sql
select country
from classes
where displacement = (select max(displacement)
    from classes)
```

(e) Find the number of ships for each country

```sql
select country, count(*)
from classes, ships
where cclass = sclass
group by country;
```

(f) Find the battle in which the number of ships sunk is highest

```sql
select battle
from
    (select battle, count(*) as scount
    from outcomes
    where result = 'sunk'
    group by battle)
where
    scount = (select max(scount)
    from (select battle, count(*) as scount
    from outcomes
    where result = 'sunk'
    group by battle));
```
3. Relational Algebra (30 pts): Specify the following queries on COMPANY database using relational algebra.

(a) Retrieve the names of all employees in department 5 who work for more than 10 hours per week on the ProductX project.
Solution:
\[
\text{employee} \leftarrow (\sigma_{\text{PNAME}=\text{ProductX}}(\text{PROJECT})) \land \sigma_{\text{PNUMBER}=\text{PNO}} \text{WORKS_ON}\]
\[
\text{employee}_{10} \leftarrow \text{employee} \land \sigma_{\text{SSN}=\text{ESSN}} (\sigma_{\text{HOURS}>10} (\text{employee} \land \text{WORKS_ON}))
\]
\[
\text{RESULT} \leftarrow \Pi_{\text{FNAME}, \text{MINIT}, \text{LNAME}} (\sigma_{\text{DNO}=5} (\text{employee}_{10}))
\]

(b) Retrieve the names of all employees who work on every project.
Solution:
\[
\text{PROJECT} \land \text{EMPL} \leftarrow \Pi_{\text{PNO}, \text{SSN}} (\text{WORKS_ON})
\]
\[
\text{PROJECTS} \leftarrow \Pi_{\text{PNUMBER}} (\text{PROJECT})
\]
\[
\text{EMP_ALL_PROJ} \leftarrow \text{PROJECT} \land \text{EMPL} \div \text{PROJECTS}
\]
\[
\text{RESULT} \leftarrow \Pi_{\text{FNAME}, \text{LNAME}} (\text{employee} \land \text{EMP_ALL_PROJ})
\]

(c) For each project, list the project name, and the total hours per week (by all employees) spent on the project.
Solution:
\[
\text{PROJECT} \land \text{HOURS} \leftarrow \Pi_{\text{PNO}, \text{TOTAL_HRS}} (\text{TOTAL_HRS})
\]
\[
\text{RESULT} \leftarrow \Pi_{\text{PNAME}, \text{TOTAL_HRS}} (\text{PROJECT} \land \text{HOURS} \land \sigma_{\text{PNO}=\text{PNUMBER}} (\text{PROJECT}))
\]

(d) Retrieve the names of all employees who do not work on any project.
Solution:
\[
\text{ALL_EMPS} \leftarrow \Pi_{\text{SSN}} (\text{employee})
\]
\[
\text{WORK_EMPS} \leftarrow \Pi_{\text{SSN}} (\text{WORK} \land \text{ON})
\]
\[
\text{NON_WORK} \leftarrow (\text{ALL_EMPS} \div \text{WORK_EMPS})
\]
\[
\text{RESULT} \leftarrow \Pi_{\text{FNAME}, \text{LNAME}} (\text{employee} \land \text{NON_WORK})
\]

(e) Retrieve the average salary of all female employees.
Solution:
\[
\text{RESULT} \leftarrow \exists_{\text{AVERAGE SALARY}} (\sigma_{\text{SEX}=\text{F}} (\text{employee}))
\]