

Solutions for Exercise Session 1 n SQL

Question 1: Basics

```
CREATE TABLE employees
( employee_number int NOT NULL,
  employee_name char(50) NOT NULL,
  department_id int,
  salary int,
  CONSTRAINT employees_pk PRIMARY KEY (employee_number),
  CONSTRAINT fk_departments
  FOREIGN KEY (department_id)
  REFERENCES departments(department_id)
);
```

Question 2: Supplier

```
SELECT S.sname
FROM Suppliers S
  WHERE S.sid NOT IN (SELECT C.sid
                     FROM Catalog C
                     WHERE C.pid NOT IN (SELECT P.pid
                                         FROM Parts P
                                         WHERE P.color <> 'blue'));
```

Question 3: Employees

```
a)Select e.empId
From employee e
  Where not exists
  (Select * From employee s where s.department = "5" and
   s.salary >=e.salary);

b)Select department, max(salary) from employee GROUP BY department;

c)select * from employee where UPPER(name) like "%JOE%";
```

Question 4: Duplicates

```
SELECT name, section FROM tbl
GROUP BY name, section
HAVING COUNT(*) > 1;
```

Question 5: Company

a)

```
select employee.employee-name, employee.street, employee.city from employee, works
where employee.employee-name=works.employee-name

and company-name = 'First Bank Corporation' and salary > 10000) ;
```

b)

```
select e.employee-name
from employee e, works w, company c
where e.employee-name = w.employee-name and e.city = c.city

and w.company-name = c.company-name ;
```

c)

```
select p.employee-name
from employee p, employee r, manages m
where p.employee-name = m.employee-name and m.manager-name = r.employee-name

and p.street = r.street and p.city = r.city ;
```

d)

```
select employee-name
from works
where salary > all (select salary

from works
where company-name = 'Small Bank Corporation') ;
```

e)

```
select company-name
from works
group by company-name
having sum(salary) <= all (select sum(salary)

from works
group by company-name) ;
```

f)

UPDATE employee

Set email = '<email id here>';

g) select e.name, m.name from manages e, manages m

where e.manager_name = m.manager_name;

Question 6:

a) SELECT temperature, heartRate
FROM Patients, Tests
WHERE pid = patient and year < 1950;

b) CREATE VIEW FreeBeds AS
SELECT num as ward, numBeds - COUNT(pid) AS numBeds
FROM Wards LEFT OUTER JOIN PatientInWard ON ward = num
GROUP BY num, numBeds
;

Question 7: Planets

a)
CREATE TABLE Planets (
star TEXT NOT NULL ,
name TEXT NOT NULL ,
distance FLOAT NOT NULL CHECK (distance > 0) ,
mass FLOAT NOT NULL CHECK (mass > 0) ,
atmosphere BOOLEAN NOT NULL ,
oxygen FLOAT NOT NULL CHECK ((oxygen = 0 and not atmosphere) OR (atmosphere AND oxygen >= 0)) ,
water FLOAT NOT NULL ,
PRIMARY KEY (star , name) ,
UNIQUE (star , distance)
);

b)
SELECT COUNT (*) FROM Planets WHERE
distance > (SELECT distance FROM Planets WHERE
star = 'Kerbol ' AND name = 'Duna ');

OR

SELECT COUNT(p1.*) FROM Planets p1, Planets p2
WHERE p1.distance > p2.distance
AND p2.star = 'Kerbol' AND p2.name = 'Duna';

c) (SELECT star , name , 'habitable ' FROM Planets WHERE
distance >= 100 AND distance <= 200 AND
atmosphere AND oxygen >= 15 AND oxygen <= 25 AND
water > 0)
UNION
(SELECT star , name , 'uninhabitable ' FROM Planets WHERE NOT(
distance >= 100 AND distance <= 200 AND

```
atmosphere AND oxygen >= 15 AND oxygen <= 25 AND  
water > 0));
```

OR

```
WITH habitables AS ( SELECT star , name FROM planets WHERE  
distance >= 100 AND distance <= 200 AND  
atmosphere AND oxygen >= 15 AND oxygen <= 25 AND  
water > 0)  
SELECT star , name , 'habitable ' FROM Planets WHERE  
(star , name ) IN ( SELECT star , name from habitables )  
UNION  
SELECT star , name , 'unhabitable ' FROM Planets WHERE  
(star , name ) NOT IN ( SELECT star , name from habitables );
```