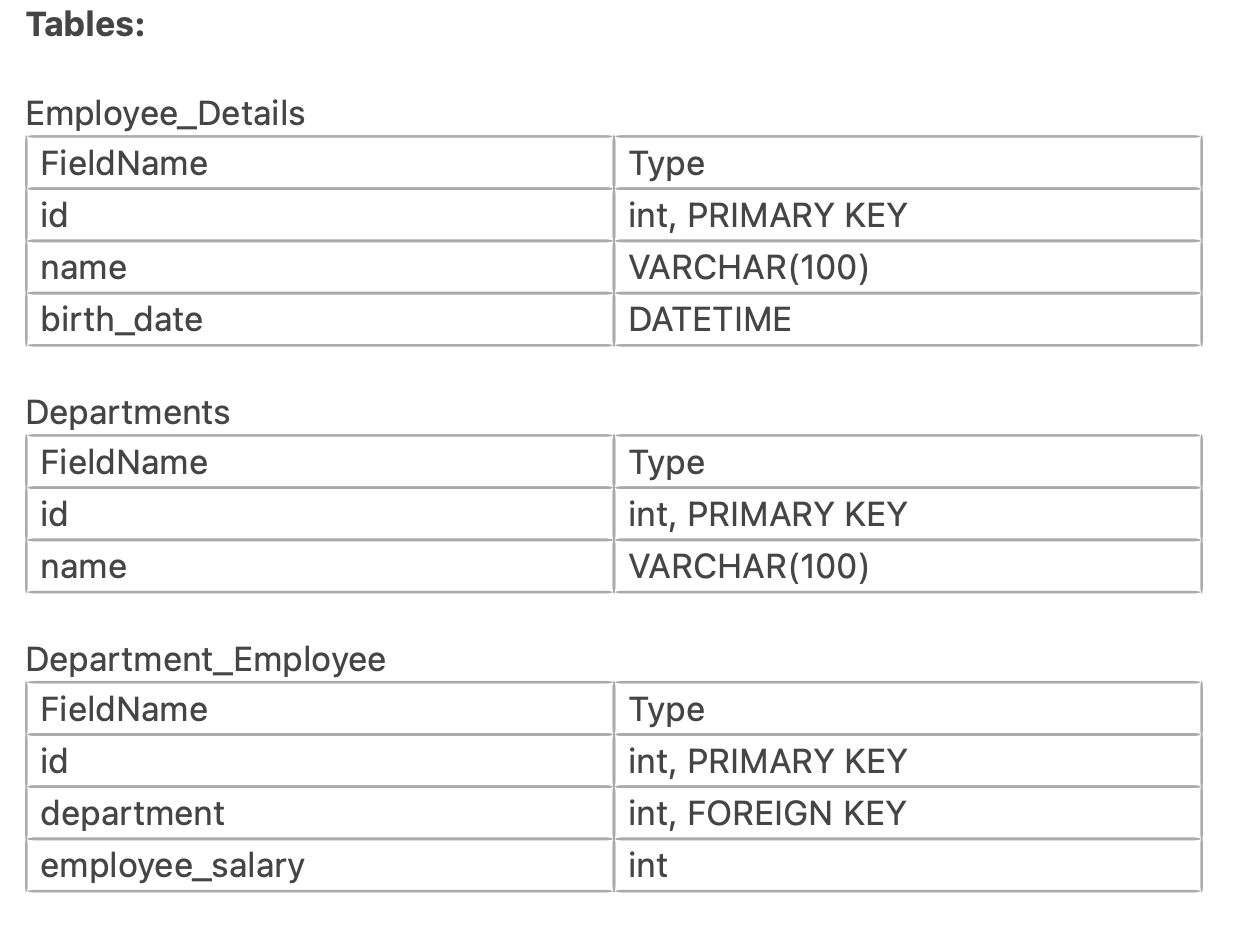
Q1. Refer to the tables below. Write SQL query to:



1. Find total number of employees

* SELECT \* FROM Employee\_Details;

1. Find total number of employees in every department with department name

* SELECT d.name, COUNT(e.id) AS total\_emp

FROM Department\_Employee e INNER JOIN Departments d

ON e.depid=d.depid

GROUP BY d.name;

1. Employees above age 30

* SELECT id, name, (YEAR(CURDATE()) - YEAR(birthdate)) AS emp\_age

FROM Employee\_Details

WHERE (YEAR(CURDATE()) - YEAR(birthdate)) >30;

1. Employees below age 25 and salary more than 30

* SELECT e.id, e.name, (YEAR(CURDATE()) - YEAR(e.birthdate)) AS emp\_age, d.salary

FROM Employee\_Details e INNER JOIN Department\_Employee d

ON e.id=d.id

WHERE (YEAR(CURDATE()) - YEAR(birthdate)) <25 AND d.salary >30;

1. Highest salary for every department

* SELECT d.name, MAX(de.salary)

FROM Departments d INNER JOIN Department\_Employee de

ON d.depid=de.depid

GROUP BY d.name;

1. 3rd highest salary for every department

* WITH rank\_sal AS

(

SELECT d.name, RANK() OVER(PARTITION BY d.name ORDER BY de.salary DESC) AS ranks

FROM Departments d INNER JOIN Department\_Employee de

ON d.depid=de.depid

)

SELECT \* FROM rank\_sal

WHERE ranks=3;

1. Find the top 2 salaried people in every department.

* WITH rank\_sal AS

(

SELECT de.id,d.name, RANK() OVER(PARTITION BY d.name ORDER BY de.salary\_in\_thousand DESC) AS ranks

FROM Departments d INNER JOIN Department\_Employee de

ON d.depid=de.depid

)

SELECT d.name,rs.ranks,rs.name

FROM Employee\_Details d INNER JOIN rank\_sal rs

ON d.id=rs.id

WHERE ranks<=2

ORDER BY rs.name,rs.ranks;

1. Find the top 2 unique salaries in every department

* SELECT COUNT(DISTINCT de.salary\_in\_thousand) AS uni\_sal,d.name

FROM dep d INNER JOIN dep\_emp de

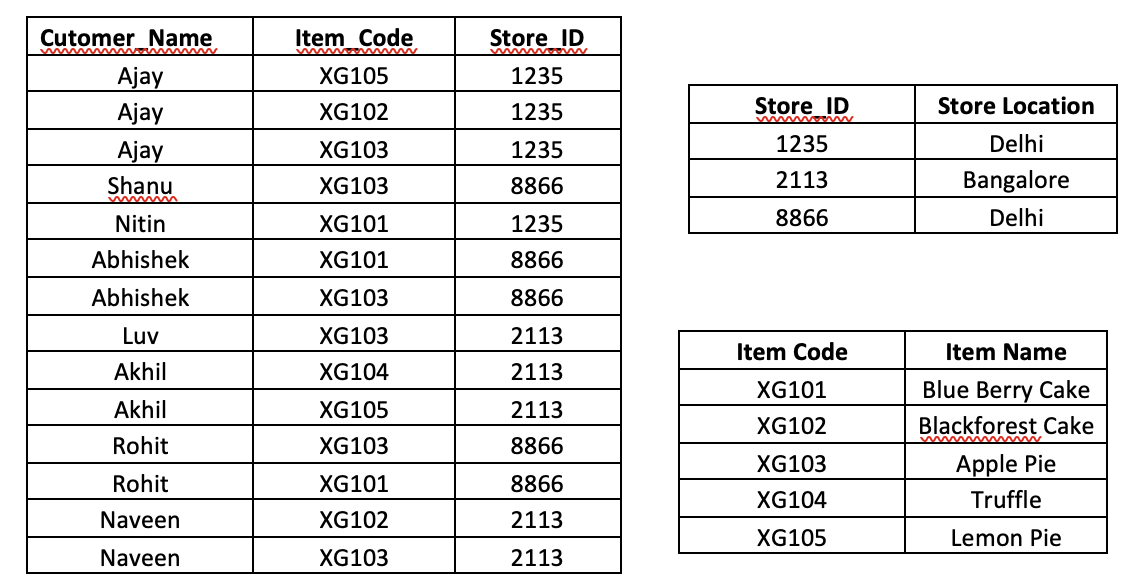
ON d.depid=de.depid

GROUP BY d.name

ORDER BY uni\_sal DESC

LIMIT 2;

Q2. Analyse 3 tables below (Customer sales ,store and product data) and write a nested query to find out the name of the most popular item in each of the 2 cities:



WITH pop\_item AS

(

SELECT c.item\_code,

s.store\_location

FROM customer\_sales c INNER JOIN store s

on c.store\_id = s.store\_id

)

SELECT p.store\_location,

COUNT(p.item\_code) AS popularproduct,

pr.item\_name

FROM pop\_item p INNER JOIN product pr

ON p.item\_code=pr.item\_code

GROUP BY pr.item\_name, p.store\_location

ORDER BY popularproduct DESC

LIMIT 4;

Q3. You are an analyst helping a Vet Clinic by making some sense of the raw data available. The data is saved across four files. [pets.csv](https://drive.google.com/file/d/1Is9M76mj6p2jzX4-LOx443reNNFfiVzD/view?usp=drive_link), [owners.csv](https://drive.google.com/file/d/1pWp4Qmdgzet_HUVdLp8YjhQ-PZZabVPT/view?usp=drive_link), [proceduredetails.csv](https://drive.google.com/file/d/1AiluhrU6DpAtqbhpsEurxk5e0Or6Q8GX/view?usp=drive_link), [procedurehistory.csv](https://drive.google.com/file/d/17LwjBW8dTZs53-Yn5oJGurMBWw4LfgbK/view?usp=drive_link). Upload all of them in your SQL Server and do the below analysis:

1. Write a query to find owner names whose pet’s name contains ‘u’ or the pet is a ‘Parrot’

* SELECT \* FROM pets

WHERE name LIKE '%u%'

OR kind='Parrot';

1. Write a query to find the cumulative sum of total procedures performed for each pet at a monthly level.

* WITH petprice AS

(

SELECT ph.petid,

ph.prdate,

pd.type,

pd.subcode,

pd.price,

SUM(pd.price) OVER (PARTITION BY ph.petid ORDER BY MONTH(ph.prdate) ASC) AS monthly\_price

FROM procedurehist ph

INNER JOIN proceduredetails pd

ON ph.type=pd.type AND ph.subcode=pd.subcode

)

SELECT p.petid, p.name, p.age, pp.prdate,pp.price,pp.monthly\_price

FROM petprice pp INNER JOIN pets p

ON pp.petid=p.petid;

1. Write a query to extract information for those pets who have undergone procedure type “Vaccinations” with sub code 3, 4 and 5 or undergone procedure type “General Surgeries” with sub code 8,10,13,15,16.

* SELECT ph.petid, p.name, p.kind, p.gender, p.age, ph.type, ph.subcode

FROM procedurehist ph

INNER JOIN pets p

ON ph.petid=p.petid

WHERE (ph. type='VACCINATIONS' AND ph.subcode = 3 OR 4 OR 5) OR (ph.type='GENERAL SURGERIES' AND ph.subcode = 2 OR 4 OR 8);

1. Write a query to find pet’s information who did not have any procedure performed

* SELECT p.\*

FROM pets p LEFT JOIN procedurehist ph

ON p.petid=ph.petid

WHERE p.petid NOT IN

(SELECT p.petid

FROM pets p INNER JOIN procedurehist ph

ON p.petid=ph.petid);

1. Write a query to find the average price incurred by each owner for their pet’s procedure

* WITH owner\_info AS

(

SELECT ph.petid, ROUND(AVG(pd.price)) AS price\_per\_pet

FROM procedurehist ph

INNER JOIN proceduredetails pd

ON ph.type=pd.type AND ph.subcode=pd.subcode

GROUP BY ph.petid

)

SELECT p.petid, o.\*,oi.price\_per\_pet

FROM pets p INNER JOIN owners o

ON p.owner\_id=o.owner\_id

INNER JOIN owner\_info oi

ON p.petid=oi.petid

WHERE p.petid IN

(SELECT petid from procedurehist);

1. Write a query to find the first and last date of procedure performed on each of the pets. The output should have columns (Pet Name, First Procedure Date, Last Procedure Date)

* SELECT p.name, MIN(ph.prdate) AS 'first procedure date' , MAX(ph.prdate) AS 'last procedure date'

FROM pets p INNER JOIN procedurehist ph

ON p.petid=ph.petid

GROUP BY p.name;

Q4. Determine the no. of records that we will get when you perform

1.Inner join

* 17

2. Left join

* 17

3. Full Outer join

* 18

4. Right join

* 18

The joining key for the two tables is Column A.

