**SQL QUESTIONS**

1. Answer each of the following questions. The questions are based on the following relational schema:

Emp(*eid:* integer, *ename:* string, *age:* integer, *salary:* real)

Works(*eid:* integer, *did:* integer, *pcttime:* integer)

Dept(*did:* integer, *dname:* string, *budget:* real, *managerid:* integer)

1. Give an example of a foreign key constraint that involves the Dept relation. What are the options for enforcing this constraint when a user attempts to delete a Dept tuple?
2. Write the SQL statements required to create the preceding relations, including appropriate versions of all primary and foreign key integrity constraints.
3. Define the Dept relation in SQL so that every department is guaranteed to have a manager.
4. Write an SQL statement to add John Doe as an employee with *eid* = 101, *age* = 32 and *salary* = 15*,* 000.
5. Write an SQL statement to give every employee a 10 percent raise.
6. Write an SQL statement to delete the Toy department.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *sid* | *name* | *login* | *age* | *gpa* |
| 53831 | Madayan | madayan@music | 11 | 1.8 |
| 53832 | Guldu | guldu@music | 12 | 2.0 |

**Figure 1.** Students with *age <* 18 on Instance *S*

Consider the SQL query whose answer is shown in Figure 1.

1. Modify this query so that only the *login* column is included in the answer.

2. If the clause WHERE *S.gpa >*= *2* is added to the original query, what is the set of tuples in the answer?

1. Suppose that we have a ternary relationship R between entity sets A, B, and C such that A has a key constraint

and total participation and B has a key constraint; these are the only constraints. A has attributes *a*1 and *a*2, with

*a*1 being the key; B and C are similar. R has no descriptive attributes. Write SQL statements that create tables

corresponding to this information so as to capture as many of the constraints as possible.

1. Briefly answer the following questions based on this schema:

Emp(*eid:* integer, *ename:* string, *age:* integer, *salary:* real)

Works(*eid:* integer, *did:* integer, *pct time:* integer)

Dept(*did:* integer, *budget:* real, *managerid:* integer)

1. Suppose you have a view SeniorEmp defined as follows:

CREATE VIEW SeniorEmp (sname, sage, salary)

AS SELECT E.ename, E.age, E.salary

FROM Emp E

WHERE E.age *>* 50

Explain what the system will do to process the following query:

SELECT S.sname

FROM SeniorEmp S

WHERE S.salary *>* 100,000

* Give an example of a view on Emp that could be automatically updated by updating Emp.
* Give an example of a view on Emp that would be impossible to update (automatically) and explain why your example presents the update problem that it does.

1. Consider the following schema:

Suppliers(*sid:* integer, *sname:* string, *address:* string)

Parts(*pid:* integer, *pname:* string, *color:* string)

Catalog(*sid:* integer, *pid:* integer, *cost:* real)

The Catalog relation lists the prices charged for parts by Suppliers. Answer the following questions:

1. Give an example of an updatable view involving one relation.
2. Give an example of an updatable view involving two relations.
3. Give an example of an insertable-into view that is updatable.
4. Give an example of an insertable-into view that is not updatable.
5. Consider the following schema:

Suppliers(*sid:* integer, *sname:* string, *address:* string)

Parts(*pid:* integer, *pname:* string, *color:* string)

Catalog(*sid:* integer, *pid:* integer, *cost:* real)

The key fields are underlined, and the domain of each field is listed after the field name. Therefore *sid* is the key for Suppliers, *pid* is the key for Parts, and *sid* and *pid* together form the key for Catalog. The Catalog relation lists the prices charged for parts by Suppliers. Write the following queries in sql

1. Find the *name*s of suppliers who supply some red part.
2. Find the *sid*s of suppliers who supply some red or green part.
3. Find the *sid*s of suppliers who supply some red part or are at 221 Packer Street.
4. Find the *sid*s of suppliers who supply some red part and some green part.
5. Find the *sid*s of suppliers who supply every part.
6. Find the *sid*s of suppliers who supply every red part.
7. Find the *sid*s of suppliers who supply every red or green part.
8. Find the *sid*s of suppliers who supply every red part or supply every green part.
9. Find pairs of *sid*s such that the supplier with the first *sid* charges more for some part than the supplier with the second *sid*.
10. Find the *pid*s of parts supplied by at least two different suppliers.
11. Find the *pid*s of the most expensive parts supplied by suppliers named YosemiteSham.
12. Find the *pid*s of parts supplied by every supplier at less than $200. (If any supplier either does not supply the part or charges more than $200 for it, the part is not selected.)
13. Consider the following relations containing airline flight information:

Flights(*flno:* integer, *from:* string, *to:* string,

*distance:* integer, *departs:* time, *arrives:* time)

Aircraft(*aid:* integer, *aname:* string, *cruisingrange:* integer)

Certified(*eid:* integer, *aid:* integer)

Employees(*eid:* integer, *ename:* string, *salary:* integer)

Note that the Employees relation describes pilots and other kinds of employees as well;every pilot is certified for some aircraft (otherwise, he or she would not qualify as a pilot), and only pilots are certified to fly. Write the following queries in SQL:

1. Find the names of aircraft such that all pilots certified to operate them have salaries more than $80,000.
2. For each pilot who is certified for more than three aircraft, find the *eid* and the maximum *cruisingrange* of the aircraft for which she or he is certified.
3. Find the names of pilots whose *salary* is less than the price of the cheapest route from Los Angeles to Honolulu.
4. For all aircraft with *cruisingrange* over 1000 miles, find the name of the aircraft and the average salary of all pilots certified for this aircraft.
5. Find the names of pilots certified for some Boeing aircraft.
6. Find the *aid*s of all aircraft that can be used on routes from Los Angeles to Chicago.
7. Identify the routes that can be piloted by every pilot who makes more than $100,000.
8. Find the *ename*s of pilots who can operate planes with *cruisingrange* greater than 3000 miles but are not certified on any Boeing aircraft.
9. A customer wants to travel from Madison to New York with no more than two changes of flight. List the choice of departure times from Madison if the customer wants to arrive in New York by 6 p.m.
10. Compute the difference between the average salary of a pilot and the average salary of all employees (including pilots).
11. Find the name and salary of every nonpilot whose salary is more than the average salary for pilots.
12. Find the names of employees who are certified only on aircrafts with cruising range longer than 1000 miles.
13. Find the names of employees who are certified only on aircrafts with cruising range longer than 1000 miles, but on at least two such aircrafts.
14. Find the names of employees who are certified only on aircrafts with cruising range longer than 1000 miles and who are certified on some Boeing aircraft.
15. Find the *eid*s of pilots certified for some Boeing aircraft.
16. Find the *name*s of pilots certified for some Boeing aircraft.
17. Find the *aid*s of all aircraft that can be used on non-stop flights from Bonn to Madras.
18. Identify the flights that can be piloted by every pilot whose salary is more than $100,000.
19. Find the names of pilots who can operate planes with a range greater than 3,000 miles but are not certified on any Boeing aircraft.
20. Find the *eid*s of employees who make the highest salary.
21. Find the *eid*s of employees who make the second highest salary.
22. Find the *eid*s of employees who are certified for the largest number of aircraft.
23. Find the *eid*s of employees who are certified for exactly three aircraft.
24. Find the total amount paid to employees as salaries.
25. Find the *aid*s of all aircraft that can be used on routes from Los Angeles to Chicago.
26. Consider the following relations:

Student(*snum:* integer, *sname:* string, *major:* string, *level:* string, *age:* integer)

Class(*name:* string, *meets at:* string, *room:* string, *fid:* integer)

Enrolled(*snum:* integer, *cname:* string)

Faculty(*fid*: integer, *fname:* string, *deptid:* integer)

The meaning of these relations is straightforward; for example, Enrolled has one record per student-class pair such that the student is enrolled in the class.Write the following queries in SQL. No duplicates should be printed in any of the answers.

1. Find the names of all Juniors (level = JR) who are enrolled in a class taught by I. Teach.
2. Find the age of the oldest student who is either a History major or enrolled in a course taught by I. Teach.
3. Find the names of all classes that either meet in room R128 or have five or more students enrolled.
4. Find the names of all students who are enrolled in two classes that meet at the same time.
5. Find the names of faculty members who teach in every room in which some class is taught.
6. Find the names of faculty members for whom the combined enrollment of the courses that they teach is less than five.
7. For each level, find the level and the average age of students for that level.
8. For all levels except JR, print the level and the average age of students for that level.
9. For each faculty member that has taught classes only in room R128, find the faculty member’s name and the total number of classes she or he has taught.
10. Find the names of students enrolled in the maximum number of classes.
11. Find the names of students not enrolled in any class.
12. For each age value that appears in Students, find the level value that appears most often. For example, if there are more FR level students aged 18 than SR, JR, or SO students aged 18, you should print the pair (18, FR).
13. Consider the following schema:

Suppliers(*sid:* integer, *sname:* string, *address:* string)

Parts(*pid:* integer, *pname:* string, *color:* string)

Catalog(*sid:* integer, *pid:* integer, *cost:* real)

The Catalog relation lists the prices charged for parts by Suppliers. Write the following queries in SQL:

1. Find the *pname*s of parts for which there is some supplier.
2. Find the *sname*s of suppliers who supply every part.
3. Find the *sname*s of suppliers who supply every red part.
4. Find the *pname*s of parts supplied by Acme Widget Suppliers and no one else.
5. Find the *sid*s of suppliers who charge more for some part than the average cost of that part (averaged over all the suppliers who supply that part).
6. For each part, find the *sname* of the supplier who charges the most for that part.
7. Find the *sid*s of suppliers who supply only red parts.
8. Find the *sid*s of suppliers who supply a red part and a green.
9. Find the *sid*s of suppliers who supply a red part or a green part.
10. For every supplier that only supplies green parts, print the name of the supplier and the total number of parts that she supplies.
11. For every supplier that supplies a green part and a red part, print the name and price of the most expensive part that she supplies.
12. Consider the following relational schema. An employee can work in more than one department; the *pct time* field of the Works relation shows the percentage of time that a given employee works in a given department.

Emp(*eid:* integer, *ename:* string, *age:* integer, *salary:* real)

Works(*eid:* integer, *did:* integer, *pct time:* integer)

Dept(*did:* integer, *dname:* string, *budget:* real, *managerid:* integer)

Write the following queries in SQL:

1. Find the names and ages of each employee who works in both the Hardware department and the Software department.
2. For each department with more than 20 full-time-equivalent employees (i.e., where the part-time and full-time employees add up to at least that many full-time employees), print the *did* together with the number of employees that work in that department.
3. Print the name of each employee whose salary exceeds the budget of all of the
4. departments that he or she works in.
5. Find the *managerid*s of managers who manage only departments with budgets greater than $1 million.
6. Find the *ename*s of managers who manage the departments with the largest budgets.
7. If a manager manages more than one department, he or she *controls* the sum of all the budgets for those departments. Find the *managerid*s of managers who control more than $5 million.
8. Find the *managerid*s of managers who control the largest amounts.
9. Find the *ename*s of managers who manage only departments with budgets larger than $1 million, but at least one department with budget less than $5 million.

|  |  |  |  |
| --- | --- | --- | --- |
| Sid | Sname | Rating | age |
| 18 | Jones | 3 | 30 |
| 41 | jonah | 6 | 56 |
| 22 | Ahab | 7 | 44 |
| 63 | moby | null | 15 |

**Figure 2.** An Instance of Sailors

Consider the instance of the Sailors relation shown in Figure 2

1. Write SQL queries to compute the average rating, using AVG; the sum of the ratings, using SUM; and the number of ratings, using COUNT.
2. If you divide the sum just computed by the count, would the result be the same as the average? How would your answer change if these steps were carried out with respect to the *age* field instead of *rating*?
3. Consider the following query: *Find the names of sailors with a higher rating than all sailors with age <* 21*.* The following two SQL queries attempt to obtain the answer to this question. Do they both compute the result? If not, explain why.

Under what conditions would they compute the same result?

SELECT S.sname

FROM Sailors S

WHERE NOT EXISTS ( SELECT \*

FROM Sailors S2

WHERE S2.age *<* 21

AND S.rating *<*= S2.rating )

SELECT \*

FROM Sailors S

WHERE S.rating *>* ANY ( SELECT S2.rating

FROM Sailors S2

WHERE S2.age *<* 21 )

1. Consider the instance of Sailors shown in Figure 2. Let us define instance S1 of Sailors to consist of the first two tuples, instance S2 to be the last two tuples, and S to be the given instance.
2. Show the left outer join of S with itself, with the join condition being *sid=sid*.
3. Show the right outer join of S with itself, with the join condition being *sid=sid*.
4. Show the full outer join of S with itself, with the join condition being *sid=sid*.
5. Show the left outer join of S1 with S2, with the join condition being *sid=sid*.
6. Show the right outer join of S1 with S2, with the join condition being *sid=sid*.
7. Show the full outer join of S1 with S2, with the join condition being *sid=sid*.
8. Consider the following relational schema and briefly answer the questions that follow:

Emp(*eid:* integer, *ename:* string, *age:* integer, *salary:* real)

Works(*eid:* integer, *did:* integer, *pct time:* integer)

Dept(*did:* integer, *budget:* real, *managerid:* integer)

1. Define a table constraint on Emp that will ensure that every employee makes at least $10,000.
2. Define a table constraint on Dept that will ensure that all managers have *age >* 30.
3. Define an assertion on Dept that will ensure that all managers have *age >* 30.
4. Compare this assertion with the equivalent table constraint. Explain which is better.
5. Write SQL statements to delete all information about employees whose salaries exceed that of the manager of one or more departments that they work in. Be sure to ensure that all the relevant integrity constraints are satisfied after your updates.
6. Consider the following relations:

Student(*snum:* integer, *sname:* string, *major:* string,

*level:* string, *age:* integer)

Class(*name:* string, *meets at:* time, *room:* string, *fid:* integer)

Enrolled(*snum:* integer, *cname:* string)

Faculty(*fid*: integer, *fname:* string, *deptid:* integer)

The meaning of these relations is straightforward; for example, Enrolled has one record per student-class pair such that the student is enrolled in the class. Write the SQL statements required to create these relations, including appropriate versions of all primary and foreign key integrity constraints. Express each of the following integrity constraints in SQL unless it is implied by the primary and foreign key constraint; if so, explain how it is implied. If the constraint cannot be expressed in SQL, say so. For each constraint, state what operations (inserts, deletes, and updates on specific relations) must be monitored to enforce the constraint.

1. Every class has a minimum enrollment of 5 students and a maximum enrollment of 30 students.
2. At least one class meets in each room.
3. Every faculty member must teach at least two courses.
4. Only faculty in the department with *deptid=33* teach more than three courses.
5. Every student must be enrolled in the course called Math101.
6. The room in which the earliest scheduled class (i.e., the class with the smallest *meets at* value) meets should not be the same as the room in which the latest scheduled class meets.
7. Two classes cannot meet in the same room at the same time.
8. The department with the most faculty members must have fewer than twice the number of faculty members in the department with the fewest faculty members.
9. No department can have more than 10 faculty members.
10. A student cannot add more than two courses at a time (i.e., in a single update).
11. The number of CS majors must be more than the number of Math majors.
12. The number of distinct courses in which CS majors are enrolled is greater than the number of distinct courses in which Math majors are enrolled.
13. The total enrollment in courses taught by faculty in the department with *deptid=33* is greater than the number of Math majors.
14. There must be at least one CS major if there are any students whatsoever.
15. Faculty members from different departments cannot teach in the same room.

Answer each of the following questions briefly. The questions are based on the following relational schema:

Emp(eid: integer, ename: string, age: integer, salary: real)

Works(eid: integer, did: integer, pct time: integer)

Dept(did: integer, dname: string, budget: real, managerid: integer)

**Set 1**

1. Write the SQL statements required to create the above relations, including appropriate versions of all primary and foreign key integrity constraints.

3. Define the Dept relation in SQL so that every department is guaranteed to have a manager.

4. Write an SQL statement to add `John Doe' as an employee with eid = 101, age = 32 and salary = 15000.

5. Write an SQL statement to give every employee a 10% raise in the salary.

6. Write an SQL statement to delete the `Toy' department. Given the referential integrity constraints you chose for this schema, explain what happens when this statement is executed.

**Set 2**

1. Define a table constraint on emp such that for any employee salary should not be less than 5000, age should not be less than 18 years. Also ensure that age should not be less than 30 for any manager.
2. List the employees which work under the manager who controls the largest amount. [A manager can be managing more than 2 departments]
3. Display the name & salary of managers who manage the departments with the top 3 largest budgets.
4. List the employee name, employee salary, manager name and the total working hours of the employee in the company.
5. List the department id, department name, its manager name and the number of employees working in that department [excluding manager].

**Set 3**

Enforce the following constraints and for each constraint, state what operations (inserts, deletes, and updates on specific relations) must be monitored to enforce the constraint.

1. Every class has a minimum enrollment of 5 students and a maximum enrollment of 30 students.
2. Every student must be enrolled in the course called Math101.
3. No department can have more than 10 faculty members.
4. Every faculty member must teach at least two courses.
5. Two classes cannot meet in the same room at the same time.

Write the following queries in SQL. No duplicates should be printed in any of the answers.

Student(snum: integer, sname: string, major: string, level: string, age: integer)

Class(cname: string, meets at: time, room: string, fid: integer)

Enrolled(snum: integer, cname: string)

Faculty (fid: integer, fname: string, deptid: integer)

**Set 1**

1. Find the names of all classes that either meet in room R128 or have five or more students enrolled.
2. Find the names of all Juniors (Level = JR) who are enrolled in a class taught by I. Teach.
3. Find the age of the oldest student who is either a History major or is enrolled in a course taught by I. Teach.
4. Find the names of all students who are enrolled in two classes that meet at the same time.

**Set 2**

1. Find the names of faculty members who teach in every room in which some class is taught.
2. Find the names of faculty members for whom the combined enrollment of the courses that they teach is less than five.
3. Print the Level and the average age of students for that Level, for all Levels except JR.
4. Find the names of students who are enrolled in the maximum number of classes.
5. Find the names of students who are not enrolled in any class.

Write the following queries in SQL for the following schema.

Suppliers(sid: integer, sname: string, address: string)

Parts(pid: integer, pname: string, color: string)

Catalog(sid: integer, pid: integer, cost: real)

**Set 1**

1. Find the pnames of parts for which there is some supplier.
2. Find the snames of suppliers who supply every part.
3. Find the snames of suppliers who supply every red part.
4. Find the pnames of parts supplied by Acme Widget Suppliers and by no one else.
5. Find the sids of suppliers who supply a red part and a green part.

**Set 2**

1. Find the sids of suppliers who charge more for some part than the average cost of that

part (averaged over all the suppliers who supply that part).

1. For each part, find the sname of the supplier who charges the most for that part.
2. Find the sids of suppliers who supply only red parts.
3. Find the sids of suppliers who supply a red part or a green part

Write the following queries in SQL.

Project(pno, proj\_name, proj\_base\_dept, proj\_mgr, topic, budget)

Manager(mid, mgr\_name, mgr\_dept, salary, age, sex)

**Set 1**

1. Find the number of male managers.
2. Find the average age of managers.
3. Find the total of the budgets for projects managed by each manager but only for managers who are in a user-specified age range.
4. List the name of the project with the lowest budget.

Write following queries in SQL.

Flights(flno: integer, from: string, to: string, distance: integer, departs: time, arrives: time, price: integer)

Aircraft(aid: integer, aname: string, cruisingrange: integer)

Certified(eid: integer, aid: integer)

Employees(eid: integer, ename: string, salary: integer)

**Set 1**

1. Find the names of aircraft such that all pilots certified to operate them earn more than 80,000.
2. For each pilot who is certified for more than three aircraft, find the eid and the maximum cruisingrange of the aircraft that he (or she) is certified for.
3. Find the names of pilots whose salary is less than the price of the cheapest route from Los Angeles to Honolulu.
4. For all aircraft with cruisingrange over 1,000 miles, find the name of the aircraft and the average salary of all pilots certified for this aircraft.
5. Find the names of pilots certified for some Boeing aircraft.
6. Find the aids of all aircraft that can be used on routes from Los Angeles to Chicago.
7. Print the enames of pilots who can operate planes with cruisingrange greater than 3000 miles, but are not certified on any Boeing aircraft.

Write the following queries in SQL.

Product (maker, model, type)

PC(mode1, speed, ram, hd, rd, price)

Laptop(mode1, speed, ram, hd, screen, price)

Printer(mode1, color, type, price)

**Set 1**

1. Find the average speed of PC's.
2. Find the average speed of laptops costing over Rs. 2000.
3. Find the average price of PC's made by manufacturer A.
4. Find the average price of PC's and laptops made by manufacturer D.
5. Find, for each different speed the average price of a PC.

**Set 2**

1. Find for each manufacturer, the average screen size of its laptops.
2. Find the manufacturers that make at least three different models of PC.
3. Find for each manufacturer who sells PC's the maximum price of a PC.
4. Find, for each speed of PC above 800, the average price.
5. Find the average hard disk size of a PC for all those manufacturers that make printers.
6. Suppose employee information is stored in following format as

Employee

|  |  |  |
| --- | --- | --- |
| EmpName | ManagerName | Salary |
|  |  |  |

1. Write a CREATE TABLE statement with constraint if every manager is also an employee.
2. Write SELECT statement to print number of employee working directly or indirectly under each manager.
3. Write SELECT statement to print name of manager having maximum number of employee working under him/her(directly or indirectly).
4. Write name of top three employees in terms of salary who are having ‘Anil’ as manager(directly or indirectly).
5. Write UPDATE statement to add Rs. 100 in salary of ‘Anil’ if both ‘Anil’ and ‘Sunil’ are having same manager (directly or indirectly).
6. Data of sales of cars is given as follows. (Sales is given at district level and covering all districts and all states)

Sales

Region Car Qty Month

Nagpur Indica 566 Jan-2011

Amravati Indica 455 Jan-2011

Nagpur Maruti 777 Jan-2011

…

Location

Region Pregion

Nagpur Vidharbha

Vidharbha Maharashtra

Maharashtra India

…

Write SQL statement for the following queries

1. Top three cars in terms of sales in Maharashtra in Jan-2011.
2. Total sales of each car in India in Jan-2011.
3. Ratio of total car sales in Jan-2011 to total car sales in Feb-2011.
4. Number of districts in India.
5. Total sales of all combination of Region, Car and Month using CUBE query.

**SQL QUESTIONS**

1. Branch (branch-name, branch-city, assets)

Customer (customer-name, customer-street, customer-city)

Loan (loan-number, branch-name, amount)

Borrower (customer-name, loan-number)

Account (account-number, branch-name, balance)

Depositor (customer-name, account-number)

Banking enterprise Database

Construct the following SQL queries for this relational database.

1. Find the names of all branches in the loan relation.
2. Find all loan numbers for loans made at the Perryridge branch with loan amounts greater than $1200.
3. Find the loan number of those loans with loan amounts between $90,000 and $100,000.
4. Find the customer names, loan numbers, and loan amounts for all customers who have a loan from the bank.
5. Find the customer names, loan numbers, and loan amounts for all loans at the Perryridge branch.
6. Find the names of all branches that have assets greater than atleast one branch located in Brooklyn.
7. Find the names of all customers whose street address includes the substring ‘Main’.
8. Find all customers who have both a loan and an account at the bank.
9. Find all customers who have an account but no loan at the bank.
10. Find the average account balance at each branch.
11. Find the number of depositors for each branch.
12. Find the average balance for each customer who lives in Harrison and has at least three accounts.
13. Find all loan numbers that appear in the loan relation with null values for amount.
14. Find all customers who have an account at all the branches located in Brooklyn.
15. Find all customers who have atmost one account at the Perryridge branch.
16. Find all customers who have at least two accounts at the Perryridge branch.
17. Find all branches where the total account deposit is less than the average of the total account deposits at all branches.
18. Delete all account tuples in the Perryridge branch.
19. Delete all account tuples at every branch located in Needham.
20. Delete the records of all accounts with balances below the average at the bank.
21. Person (driver-id#, name, address)

Car (license, model, year)

Accident (report-number, date, location)

Owns (driver-id#, license)

Participated (driver-id, car, report-number, damage-amount)

Insurance database

Consider the above insurance database, where the primary keys are underlined. Construct the following SQL queries for this relational database.

a. Find the total number of people who owned cars that were involved in accidents in 1989.

b. Find the number of accidents in which the cars belonging to “John Smith” were involved.

c. Add a new accident to the database; assume any values for required attributes.

d. Delete the Mazda belonging to “John Smith”.

e. Update the damage amount for the car with license number“AABB2000” in the accident with

report number“AR2197” to $3000.

1. Employee(employee-name, street, city)

Works (employee-name, company-name, salary)

Company (company-name, city)

Manages (employee-name, manager-name)

Employee database.

Consider the above employee database, where the primary keys are underlined.

Give an expression in SQL for each of the following queries.

1. Find the names of all employees who work for First Bank Corporation.
2. Find the names and cities of residence of all employees who work for First

Bank Corporation.

1. Find the names, street addresses, and cities of residence of all employees who work for First Bank Corporation and earn more than $10,000.
2. Find all employees in the database who live in the same cities as the companies for which they work.
3. Find all employees in the database who live in the same cities and on the same streets as do their managers.
4. Find all employees in the database who do not work for First Bank Corporation.
5. Find all employees in the database who earn more than each employee of Small Bank Corporation.
6. Assume that the companies may be located in several cities. Find all companies located in every city in which Small Bank Corporation is located.
7. Find all employees who earn more than the average salary of all employees of their company.
8. Find the company that has the most employees.
9. Find the company that has the smallest payroll.
10. Find those companies whose employees earn a higher salary, on average, than the average salary at First Bank Corporation.
11. Choose an appropriate domain for each attribute and an appropriate primary key for each relation schema.
12. Write check conditions for the schema you deﬁned in above question to ensure that:
    1. Every employee works for a company located in the same city as the city in which the employee lives.
    2. No employee earns a salary higher than that of his manager.
13. Using SQL, deﬁne a view consisting of manager-name and the average salary of all

employees who work for that manager.

1. Consider the above employee database. Give an expression in SQL for each of the following queries.
   1. Modify the database so that Jones now lives in Newtown.
   2. Give all employees of First Bank Corporation a 10 percent raise.
   3. Give all managers of First Bank Corporation a 10 percent raise.
   4. Give all managers of First Bank Corporation a 10 percent raise unless the salary becomes greater than $100,000; in such cases, give only a 3 percent raise.
   5. Delete all tuples in the works relation for employees of Small Bank Corporation.
2. Write an SQL query, without using a with clause, to ﬁnd all branches where the total account deposit is less than the average total account deposit at all branches,
   1. Using a nested query in the from clauser.
   2. Using a nested query in a having clause.
3. Suppose that we have a relation marks(student-id, score) and we wish to assign grades to students based on the score as follows: grade F if score<40, grade C if 40 ≤score<60, grade B if 60≤score<80, and grade A if 80≤score.Write SQL queries to do the following:
4. Display the grade for each student, based on the marks relation.
5. Find the number of students with each grade.