

Practice Questions – SQL (Queries, Nested Queries and Triggers)

1. 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, *budget*: real, *managerid*: integer)

Write the following queries in SQL:

- i. Print the names and ages of each employee who works in both the Hardware department and the Software department.
 - ii. Find the **managerids** of managers who manage only departments with budgets greater than \$1,000,000.
 - iii. Find the **enames** of managers who manage the departments with the largest budget.
 - iv. If a manager manages more than one department, he or she **controls** the sum of all the budgets for those departments. Find the **managerid** of managers who control more than \$5,000,000.
 - v. Create a trigger in T-SQL to ensure that an employee does not work more than 100% in total. The attribute *pct_time* in Works relation contains the percentage that an employee works in a department.
2. A car dealer database is defined by the relational schema,

Salesman (EmpCode, Name, Street, City, PostCode, Salary)

Customer (CustomerNum, Name, Street, City, PostCode, TelNumber)

Sales (SaleNum, EmpCode, CustomerNum, CarRegNum, Date, Value)

Stock (CarRegNum, DateOfPurchase, Mileage, Colour, Manufacturer, Model, PurchasePrice)

- i. Find the cost of the purchase of the vehicle with registration number G777 XLA.
 - ii. Find the name of the salesman who sold the vehicle with registration number G777 XLA.
 - iii. Find the colour of the vehicles sold in September 1993.
 - iv. Find the number of the BMWs sold by the dealer.
 - v. Find the total sale value for each model in December 1998.

3. A retail banking organisation is going to develop a relational database using the following draft schema (the primary keys are underlined),

Account(AccCode, Balance)

Customer (CustCode, Name, Street, Number, City, Postcode, TelNumber)

Ownership(AccCode, CustCode)

SavingAccount (AccCode, IntRate)

CurrentAccount(AccCode, OverdraftLimit)

- For every tuple in the relations *Ownership*, *SavingAccount* and *CurrentAccount* there must be a tuple in the relation *Account* having the same account code with it.
- Each account is owned by a customer whose details are fully known to the system at any instance of time.
 - i. Print all the customer names and addresses who has negative balance.
 - ii. Calculate the interest amount of the customer whose customer number is 'A000004'
 - iii. Print customers who owns both current accounts and Savings account.
 - iv. Print the number of customers in each post code.
 - v. Select the postcodes that has customers less than 100.

4. A university database is defined by the relational schema,

Student (StudNum, Name, Department, CourseCode)

Module (ModNum, Title, Syllabus, LecturerNum)

Exams (StudNum, ModNum, Date, Result)

AcademicStaff (StaffNum, Name, Department)

Course (CourseCode, Title, Department)

CourseModules (CourseCode, ModNum)

- i. Find the name of the lecturer who teaches the Database Modelling module.
- ii. Find the student number of each student who is doing the degree with course code CS500 and who has passed the exams for all the modules in this degree (an exam result is considered to be a "pass" if it is greater than or equal to 50).
- iii. Find the titles of the courses offered by the department of computing.
- iv. Find the average exam mark of the students doing a degree in "software engineering".

5. Consider the following relational database schema:

DEPT(dname, location)
STUDENT(name, regno, gpa, level, dept)
COURSE(cno, cname, dept)
TAKE(regno, cno)

- i. Print the names of the courses taken by any student who is from a department located in 'Evans Hall'.
- ii. Print the names of the departments which have no students with gpa better than 3.5 taking less than 4 courses.