Xi’an Jiaotong-Liverpool University

西交利物浦大学

PAPER CODE	EXAMINER	DEPARTMENT	TEL
CPT103	Jianjun Chen	Computer Science and Software Engineering	0512 81889137

MOCK EXAMINATION

Undergraduate – Year 2

Introduction to Database Systems

TIME ALLOWED: 2 Hours

INSTRUCTIONS TO CANDIDATES

1、 This is a closed book examination.

2、 Total marks available are 100. This will count for 60% in the final assessment.

3、 Answer all questions.

4、 Answer should be written in the answer booklet(s) provided.

5、 Only English solutions are accepted.

6、 The university approved calculator - Casio FS82ES/83ES can be used.

7、 All materials must be returned to the exam supervisor upon completion of the exam. Failure to do so will be deemed academic misconduct and will be dealt with accordingly.

Question A: Basic Relational Algebra and SQL

Consider the following relations:

Students

ID	FName	LName	Year	AvgMark
001	Kaia	Harrison	3	55
002	Emma	Davis	2	77
003	Brynn	Woods	1	81
004	Cameron	Cook	1	51
005	Adam	Cunningham	2	63
006	James	Cook	1	43
007	Andrew	Middleton	3	78

Modules

ID	Name	Year	Lecturer
001	Unix Systems	1	Zoe
002	Database	2	Brett
003	C++	2	Zach
004	Compilers	3	Brett
005	Computer Graphics	2	Ted

a) You are given three SELECT queries. What are the results of application of these queries to the tables "Students" and "Modules"? Provide the answer in a table format. In case that query is not valid, explain the reason.

```
1. SELECT LName, Year FROM Students
        WHERE Year = 1 OR Year = 3

2. SELECT DISTINCT Year
        FROM Students NATURAL JOIN Modules
```

b) Assuming that all modules are mandatory, produce a list of all students (containing their IDs, first and last names) taught by Brett from the "Students" and "Modules" tables. Write your answer using a single SQL statement.

c) Write a SQL query to get the second largest AvgMark from the "Student" table. You are NOT allowed to use row_number(), rownum or rank() or limit.

d) Write a SQL query to find the number of instances of each name in the following table, where the number of instances is greater than 1. Names that appear only once (such as "d", "e" and "f") should not be included in the result.

NameList

name
a
a
b
b
b
c
c
d
e
f
gg
gg

Answers for A:

a.1

LName	Year
Harrison	3
Woods	1
Cook	1
Cook	1
Middleton	3

a.2

Year
2

The natural join returns "id = 002 and year =2", "id = 005 and year = 2".

After selecting the year attribute, the duplicated 2 is removed.

b

SQL:

```
SELECT Students.ID, Students.FName, Students.LName FROM
  Students CROSS JOIN Modules
  WHERE Students.Year = Modules.Year AND Lecturer="Brett";
```

c

```
SELECT MAX(AvgMark) FROM Students WHERE AvgMark < (SELECT
MAX(AvgMark) FROM Students)
```

d

```
SELECT NameList.name, Count(NameList.name) AS CountOfsName

     FROM NameList GROUP BY NameList.name

     HAVING (((Count(NameList.name))>1));
```

Question B: Functional Dependencies

The relation below stores information about the coaches, passengers and journeys. Assume a route from place A to place B is served by several dedicated coaches. Each coach is assigned to a fixed driver and the driver only work on that coach. RouteID identifies a travel route from place A to place B by a certain driver. Coaches are dispatched from either end of a route according to some schedules.

From the given table data:

1. Identify the Primary Key for the relation.
2. Determine all functional dependencies.
3. From these dependencies, identify which are partial (if any) and which transitive (if any).

RouteID	Driver	DepartTime	From	Destination	PassengerName	TicketNo	SeatNo

Answers for B:

***NOTE: This is not the only solution.**

PK is TicketNo

FDs:

TicketNo -> RouteID, Driver, DepartTime, From, Destination, PassengerName, SeatNo

RouteID-> Driver, From, Destination

Driver -> From, Destination

No partial dependencies.

The second and the third FDs are transitive.

Question C: Normalisation

Normalise the following table "T" into the 3rd Normal Form by clearly describing the normalisation process, i.e. the dependencies removed and how the table is split into sub-tables. Describe the functional dependencies for each resulting sub-tables.

A	B	C	D	E	F	G

Attributes (A, B, C) form the primary key and the functional dependencies:

A, B, C -> D, E, F, G

B -> D

E -> F

G -> A

Answers for C:
D is a non-key attribute that is partially dependent on the primary key.
To remove this partial dependency, we split the table into:

Table 1: [A, B, C, E, F, G]
Table 2: [B, D]

Table 1 has the following FDs:
A, B, C -> E, F, G
E -> F
G -> A

F is transitively dependent on (A, B, C) via E
To remove this transitive dependency, we split table 1 into:
Table 1-1: [A, B, C, E, G]
Table 1-2: [E, F]

Functional dependency G -> A is neither a partial nor a transitive dependency.

Question D: Entity-relationship Modelling

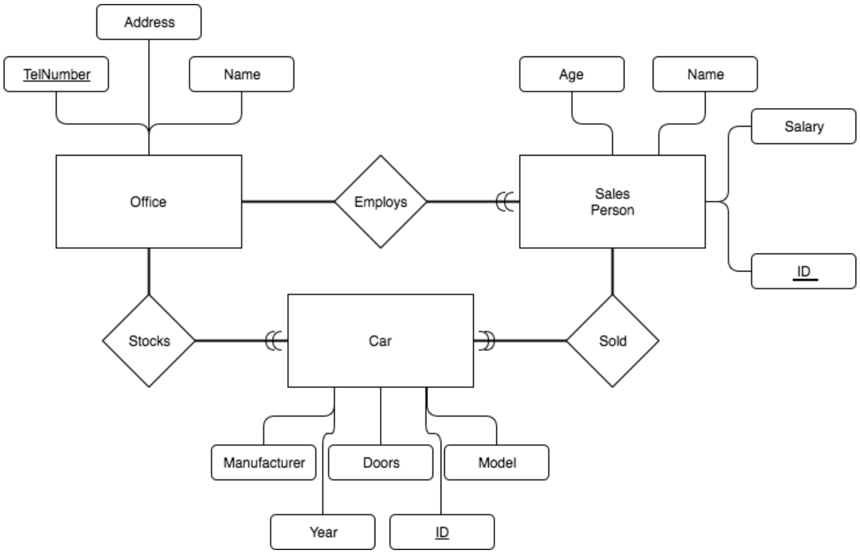
You have been hired by a large, multi-branch car sales company to design a database capable of storing the company's information. You are asked to develop a database system capable of storing the company's Offices, Inventory and Staff information. The requirements for the database are provided below:

- There are several Offices in different locations across the United Kingdom.
- Each Office has its own Name, Address and Telephone number.
- Each Office will have a unique Telephone number.
- Each Office employs many sales-persons. A sales-person can only be employed by a single Office.
- A sales-person has a Name, Age, Salary and a unique ID number.
- Each Office will have many cars associated to it. A car may only be associated to a single Office.
- Information relating to each Car is also stored. This information includes: Manufacturer, Model, Production Year and Number of Doors and a unique ID number.
- A car may be sold by a single sales-person. A sales-person may sell many cars.

Task 1: Draw the entity relationship diagram for the car sales company's database.

Task 2: Based on your solution to Task 1 above, write the SQL code to create the tables for the database. You should include all the specified attributes and specify the appropriate primary and foreign keys. Minor syntactical errors in your SQL code will not be penalised in the marking of this answer.

Answers for D task 1



Answers for D task 2:

```
CREATE TABLE Office(
      TelNumber VARCHAR(15) NOT NULL PRIMARY KEY,
      Address VARCHAR(255),
      Name VARCHAR(50)
);
```

```
CREATE TABLE SalesPerson(
      ID INT NOT NULL PRIMARY KEY,
      Name VARCHAR(50),
      Age INT,
      Salary INT,
      OfficeId VARCHAR(15),
      CONSTRAINT fkSalesToOffice FOREIGN KEY (OfficeId)
            REFERENCES Office(TelNumber)
);
```

```
CREATE TABLE Car(
      ID INT NOT NULL PRIMARY KEY,
      Model VARCHAR(50),
      Doors INT,
      Year INT,
      Manufacturer VARCHAR(50),
      OfficeId VARCHAR(15),
      SellerID INT,
      CONSTRAINT fkCarsToOffice FOREIGN KEY (OfficeId)
            REFERENCES Office(TelNumber),
      CONSTRAINT fkCarsToSales FOREIGN KEY (SellerID)
            REFERENCES SalesPerson(ID)
);
```