

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

MOCK EXAMINATION 1

TIME ALLOWED: 2 Hours

INSTRUCTIONS TO CANDIDATES

- 1、 This is a closed book examination.
- 2、 Total marks available are 100.
- 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: SQL

Consider the following relations:

Company

cID	cName	Est
1	Apple	1976
2	Microsoft	1975
3	Oracle	1977
4	Kingsoft	1988
5	Redhat	1993
6	Nintendo	1889
7	Sun Microsystems	1982

Product

pName	cID
WPS	4
Office 365	2
Apple II	1
Java	3
Switch	6
Java	7
Bug 10	2

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

1) **SELECT** cName **FROM** Company **WHERE** Est > 1990

2) **SELECT** * **FROM** Company, Product
WHERE Company.cID = Product.cID
AND (cName = 'Apple' **OR** pName = 'Java')

3) **SELECT** * **FROM** Company
WHERE cID **NOT IN** (**SELECT** * **FROM** Product)

- b) Write an SQL statement to find all companies that do not have their products listed. List company names only.
- c) Write an SQL statement to find the number of products of each company. List company names and the number of products.
- d) Write an SQL statement to list the product that is made by the oldest company.

Answers:

Q a.1

cName
Redhat

Q a.2

cID	cName	Est	pName	cID
1	Apple	1976	Apple II	1
3	Oracle	1977	Java	3
7	Sun Microsystems	1982	Java	7

Q a.3

In correct subquery. Should not return multiple column.

Q b

select c.cName

from company c

where cID not in (select cID from product p);

Q c

select c.cName, count(pName) as numProd

from company c, product p

where c.cID = p.cID

group by c.cID;

Q d

select pName

from company c, product p

where c.cID = p.cID and c.Est <= all(select Est from company);

Question B: Functional Dependencies

The relation below stores information about students, accommodation and academic advisers. Assume that more than one student may live in one room and each academic adviser advises a unique set of students. Electricity bills are applied to individual rooms.

ElectricityBill	AdviserName	StudName	RoomNo	StudEmail	AdviserID
-----------------	-------------	----------	--------	-----------	-----------

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).

Answer:

1. StudEmail is the primary key.
2. Functional dependencies:
 1. StudEmail \rightarrow All other attributes
 2. RoomNo \rightarrow ElectricityBill
 3. AdviserID \rightarrow AdviserName
3. No partial dependency, Both 2 and 3 are transitive dependencies.

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-table.

A	B	C	D	E	F	G	H
---	---	---	---	---	---	---	---

Attributes (A, B) form the primary key. The relation has the following functional dependencies:

$A, B \rightarrow C, D, E, F, G, H$

$B \rightarrow C$

$D \rightarrow E$

$E \rightarrow A$

$F \rightarrow G, H$

Answer:

C is non-key attribute that is partially dependent on primary key.

To remove this partial dependency, we split the table into:

Table 1: [A, B, D, E, F, G, H] <- Indicate the primary key with underline here.

Table 2: [B, C]

E is transitively dependent on (A, B) via D.

To remove this transitive dependency, we split table 1 into:

Table 1-1: [A, B, D, F, G, H]

Table 1-2: [D, E]

G, H is transitively dependent on (A, B) via F.

To remove this transitive dependency, we split table 1-1 into:

Table 1-1-1: [A, B, D, F]

Table 1-1-2: [F, G, H]

The final database looks like:

Table 1-1-1: [A, B, D, F]

Table 1-1-2: [F, G, H]

Table 1-2: [D, E]

Table 2: [B, C]

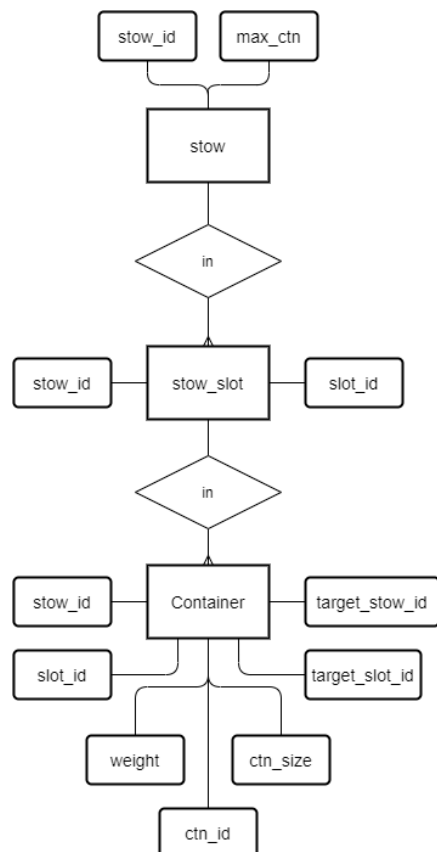
Question D: Entity-relationship Modelling

You are hired by a port company to develop a database for managing container and yard data. In this port, containers are placed in different container stows (stow = container stacking area). Each stow has an identifier, a maximum number of containers supported. A stow has several slots for containers, each slot is associated with a slot id. Each container has its container identifier, current stow, current slot in the stow, weight and size. A container transport task involves transporting one or two containers from their initial position to a same target position. The target position is described using a target stow identifier and a target stow position.

Task 1: Draw the entity relationship diagram for the port 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.

Answer:



```
create table stow (  
    stow_id int primary key,  
    max_ctn int  
);  
  
create table stow_slot (  
    stow_id int,  
    slot_id int,  
    constraint stow_slot_pk primary key (stow_id, slot_id),  
    constraint stow_slot_fk foreign key (stow_id) references stow (stow_id)  
);  
  
create table container (  
    ctn_id int primary key,  
    stow_id int,  
    slot_id int,  
    weight int,  
    ctn_size varchar(20), -- can be int  
    target_stow_id int,  
    target_slot_id int,  
    constraint container_cur_pos_fk foreign key  
        (stow_id, slot_id) references stow_slot(stow_id, slot_id),  
    constraint container_tgt_pos_fk foreign key  
        (target_stow_id, target_slot_id) references stow_slot(stow_id, slot_id)  
);
```

END OF RESIT EXAM