

The University of Jordan

King Abdullah II School for Information Technology
 Department of Computer Information Systems
 1902321: Database Management Systems.

Duration 2hrs
 Spring, 2011
 Final Exam

Instructors: Dr. Raja Alomari (coordinator), Dr. Omar Adwan, and Ms Walaa Qutechat.

Name (Print CLEARLY):..... StID:.....

Q1: (15 pts) Study the following form for a trading company that sells wholesale grocery items. Based on this form, create a database in the 3rd normal form by following the normalization rules step-by-step. [Hint, consider the SaleID as your primary key].

SALE FORM				
Sale ID:	786540-02027			
Purchase Date:	1/2/1973			
Buyer Information			Clerk information	
Buyer ID:	54298	Clerk ID:	5729	
Buyer Name:	Ahmad Hamdan	Clerk Name:	Salem Saleem	
Items information				
Item ID	Item Description	Unit Price	No of boxes	Sub-total
0301	Coffee	20	9	180
0012	Tea	30	10	300
0032	liquid Milk	2	30	60
Total				JD 1080

Step 0: Plain Table with primary key

SaleID, PurchaseDate, BuyerID, BuyerName, clerkID, clerkName, ItemID, ItemDescription, UnitPrice, NoBoxes

(5 pts) Step1: 1st Normal Form: Show all tables (Hint: Remove repeating groups if exist)

Table 1: SaleID, PurchaseDate, BuyerID, BuyerName, clerkID, clerkName,

Table 2: SaleID, ItemID, ItemDescription, UnitPrice, NoBoxes

-1 pnt per
mistake upto -3
pnts

(5 pts) Step 2: 2nd Normal Form: Show all tables (Hint: Remove partial dependencies if exist).

Table 1: <u>SaleID</u> , PurchaseDate, BuyerID, BuyerName, clerkID, clearkName,	
Table 2: <u>SaleID</u> , <u>ItemID</u> , UnitPrice, NoBoxes	FK(ItemID)
Table 3: <u>ItemID</u> , ItemDescription	

-1 pnt per mistake
upto -3 points

(5 pts) Step 3: 3rd Normal Form: Show all tables (Hint: Remove transitive dependencies if exist).

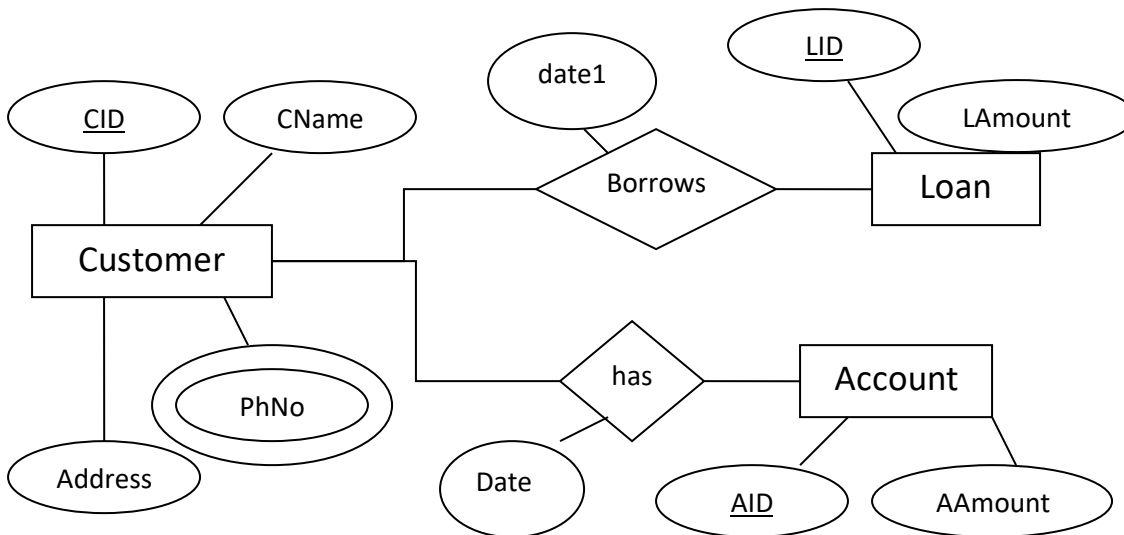
Table 1: <u>SaleID</u> , PurchaseDate, BuyerID, clerkID,	FK(BuyerID), FK(clerkID)
Table 2: <u>SaleID</u> , <u>ItemID</u> , UnitPrice, NoBoxes	FK(ItemID)
Table 3: <u>ItemID</u> , ItemDescription	
Table 4: <u>BuyerID</u> , BuyerName	
Table 5: <u>clerkID</u> , clerkName,	

-1 pnt per
mistake upto -
2pnts

Q2: (5 pts) Draw the ERM that represents the database of Q2.

Proceed to Page 3

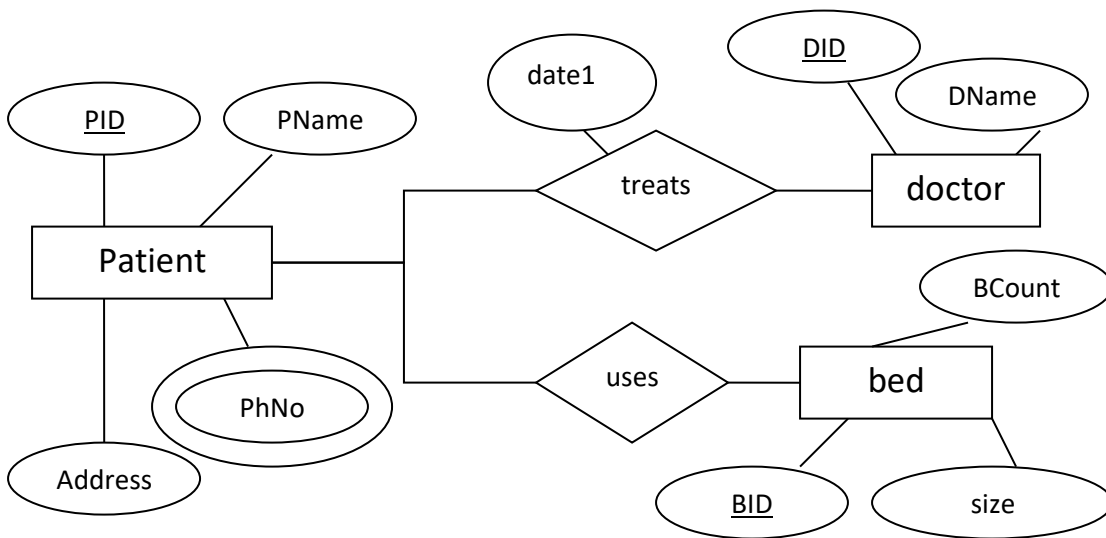
Q3: (10 pts) Study the following ERM and answer questions below (2pts each)



- The SQL statement that retrieves all customer names (**without** duplication) who had a loan is:
 - select distinct **custName** from **customer, borrow**;
 - select distinct **custName** from **customer** where **LID** in (select **LID** from **loan**);
 - c. select distinct **custName** from **customer** where **CID** in (select **CID** from **borrow**);
 - d. a + c
- The SQL statement that retrieves all customer names with accounts having an amount over JD50000 (**without** duplication) is:
 - select distinct **custName** from **customer, account** where **AAmount** > 50000;
 - b. select distinct **custName** from **customer** where **CID** in (select **CID** from **has, account** where **has.AID = account.AID** and **AAmount** > 50000);
 - c. select distinct **custName** from **customer, has** where **AAmount** > 50000;
 - d. b + c
- The SQL statement that retrieves the **summation** of all borrowed loans is:
 - select **LAmount** from **loan** where **LID** in (select **LID** from **borrow**);
 - select **sum, LAmount** from **loan, borrow** where **loan.LID = borrow.LID**;
 - select **sum, LAmount** from **loan** where **LID** in (select **LID** from **borrow**);
 - d. None of the above.
- The SQL statement that retrieves all loan transactions occurred on April 1st, 2009:
 - select * from **loan** where **date1 = '01-April-2009'**;
 - select **L.LID, L.LAmount, B.date1** from **loan L, borrow B** where **L.LID = B.LID** and **B.date1 = '01-April-2009'**;
 - select **L.LID, L.LAmount** from **loan** as L where **LID** in (select **LID** from **borrow** where **date1 = '01-April-2009'**);
 - d. b + c
- The SQL statement that retrieves customer IDs, average loan amounts grouped by the CIDs:
 - select **CID, CName** from **customer C, loan L, borrow B** where **C.CID = B.CID** and **B.LID = L.LID** and **L.LAmount = avg(L.LAmount)**;
 - b. select **B.CID, avg(L.LAmount)** from **loan L, borrow B** where **B.LID=L.LID** group by **B.CID**;
 - c. select **CID, CName** from **customer C** where **CID** in (select **CID, avg(L.LAmount)** from **borrow B, loan L** where **B.LID = L.LID** group by **B.CID**);
 - d. None of the above

Proceed to Page 4

Q4: (10 pnts) Given this ERM, Answer the following (1 pnt each):



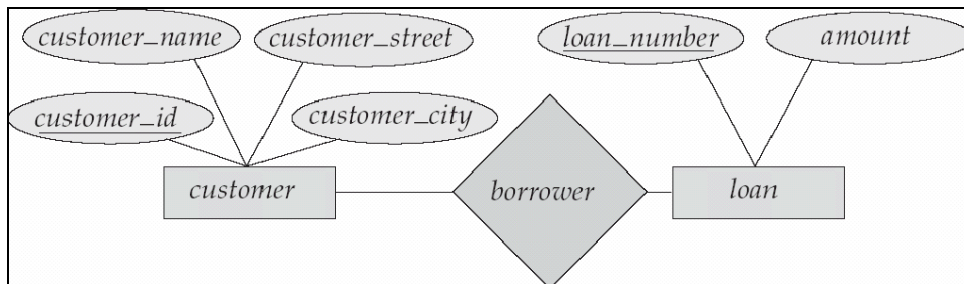
- The SQL statement " **select P.PName, Ph.PhNo from Patient P, PatientPhones Ph where P.PID = Ph.PID;**" retrieves:
 - All patient information and their phone numbers;
 - Patient names and and only one phone number.
 - All Patient names and all phone numbers.
 - This statement is wrong.
- The SQL statement " **(select P.PID from Patient P) union (select T.PID from treats T);**" retrieves:
 - All patient IDs regardless of receiving a treatment or not.
 - All patients who have been treated by a doctor.
 - Some of the patients who have been treated by a doctor.
 - None of the above.
- The SQL statement " **select P.PName from Patient P where P.PID in (select U.PID from Uses U) intersect (select T.PID from treats T);**" retrieves:
 - All patient names who received treatment and used a bed.
 - All patient names who used a bed but did not receive a treatment.
 - All patient names who received a treatment but did not use a bed.
 - None of the above.
- To create the table **uses**:
 - Create table uses (PID varchar2(20), BID varchar2(20), primary key(BID, PID));
 - Create table uses (PName varchar2(20), BID varchar2(20), primary key(BID));
 - Create table uses (PID varchar2(20), BID varchar2(20), primary key(PID,BID), foreign key(PID) references (PID), foreign key(BID) references (BID));
 - None is correct.
- To create the table **doctor**:
 - Create table doctor (DID varchar2(20), Dname varchar2(50), primary key(DID));
 - Create table doctor (DID varchar2(20), Dname varchar2(50) not null, primary key(DID), foreign key (DID) references treats(DID));
 - Create table doctor (DID varchar2(20), Dname varchar2(50) not null, primary key(DID));
 - a + c are both correct.

Proceed to Page 5

6. The statement "**update bed set size = 'queen' "** will
 - a. Add an attribute size to the table **bed** and set the default value to 'queen'.
 - b. Set the default value for any new inserted **bed** record to 'queen'
 - c.** Set all the size values for each record in the table **bed** to 'queen'
 - d. b + c are both correct
7. The statement "**insert into patient(PID, Address, PName) values ('0001','Amman', 'Ahmad') "** will
 - a. Add three attributes to the table patient.
 - b. Add one record to the table patient with an ID = '0001' but leaves both the address and PName empty because they are not in order.
 - c.** Add one record to the table patient with an ID = '0001', address = 'Amman', Name = 'Ahmad'
 - d. Either b or c might happen depending on the SQL version you are using.
8. The statement "**drop table bed where BID = '0001' "** will:
 - a. Drop all the record with ID = '0001'
 - b. Drop all the database
 - c. Drop the tables bed and uses because they depend on each other.
 - d.** This statement is wrong in syntax.
9. The statement "**update bed set BCount = case when size='queen' then BCount + 20 when size='king' then BCount+5 else BCount + 17 end; "** will:
 - a.** Nothing will happen because this statement is wrong in syntax.
 - b. The BCount value will change by adding 17 to each record because of the else clause.
 - c.** The BCount value will add 20 for all queen bed size, 5 for all king size and 17 for the other sizes.
 - d. The BCount value will add 20 for all queen bed size, (20+5) for all king size and (20+5+17) for the other sizes.
10. Given that this databases is created correctly in the database with all foreign keys and primary keys, which of the following will execute correctly:
 - a. Delete table patient;
 - b.** Drop table treats;
 - c. Remove table doctor;
 - d. All of the above will execute and the tables will no longer be in the database.

Both a and c are correct because the word 'end was not there.'

Q5: (10 pnts) Study this ERM and answer the following based on your relational algebra knowledge:



Question (2pnts each)	<u>Your Answer</u>
<p>Write a relational algebra statement to find the loan numbers (loan_number) for the loan having an amount less than 500. [Hint: use both projection (Π) and selection (σ) operators].</p>	$\Pi_{\text{loan_number}} (\sigma_{\text{amount} < 500} (\text{loan}))$
<p>Write a relational algebra statement to find the sum of loan amounts in the loan relation. [Hint: use the aggregate function (g) operator].</p>	$g_{\text{sum}(\text{amount})} (\text{loan})$
<p>Write the outcome of this expression Customer_city $g_{\text{count}(\text{customer_id})}$ (Customer)</p>	<p>Number of customers grouped by their cities.</p>
<p>Write a relational algebra statement to find all customer names, their loan IDs, and the amounts. [Hint: use projection (Π) and product (\times) operators].</p>	$\Pi_{\text{customer_name}, \text{loan_number}, \text{amount}} ((\text{customer} \times \text{borrower}) \times \text{loan})$
<p>Write a relational algebra statement to find the intersection between Customer IDs in both Customer and borrow relations. [Hint: use projection (Π) and intersection (\cap) operators].</p>	$\Pi_{\text{customer_id}}(\text{borrower}) \cap \Pi_{\text{customer_id}}(\text{customer})$ $\Pi_{\text{customer_id}}(\text{borrower} \cap \text{customer})$

End of Exam

Good Luck!

Or any
acceptable
form: 2 pnts, 1
pnts, and 0 pnt.