**Name**

Given the following Bachman Diagram and previously stored data, answer the following questions:

CUSTOMER EMPLOYEE INVENTORY

CNO CNAME CTYPE EMPNO ENAME SNO DESCR QOH LCOST LPRICE

pk pk pk

^ ^ ^

INVOICE

INO EMPNO CNO CDATE

pk sk sk

^ ^

ITEMS

INO SNO QSHIP SCOST SPRICE

pk

^ sk

^

**CUSTOMER EMPLOYEE INVENTORY**

**CNO CNAME CTYPE EMPNO ENAME SNO DESCR QOH LCOST LPRICE**

**1 Packy R 99 Joe C-144 Pantyhose 10 1.00 1.99**

**2 Patty W 100 Linda X-210 Handgernades 12 5.00 14.50**

**3 Kim W Y-333 Shot&Beer 144 4.00 6.00**

**INVOICE**

**INO EMPNO CNO IDATE**

**1000 99 2 05/01/2005**

**1001 100 2 05/02/2005**

**1002 99 3 05/02/2005**

**ITEMS**

**INO SNO QSHIP SCOST SPRICE**

**1000 C-144 4 1.00 1.50**

**1000 X-210 5 4.50 13.95**

**1001 C-114 1 1.00 1.99**

**1002 X-210 10 5.00 14.50**

**A schema is a collection of database objects (as far as this hour is concerned—tables) associated with one particular database username. This username is called the schema owner, or the owner of the related group of objects. Given JPLST1.PACKINGLIST, the schema or owner name is JPLST1, then followed by a period, then followed the table or column name.**

1. **Code an appropriate CREATE TABLE CUSTOMER which illustrates the design of any Primary Key, Check or Foreign Key Constraints, and an appropriate CREATE INDEX statement that could have been used to for the CUSTOMER table. The CUSTOMER TYPE must be 'W','R','O', which indicates wholesale, retail and other. You must prefix your table name by your schema name, or table owner name (7 points).**

**CREATE TABLE PXGST128.CUSTOMER**

**(CUSTOMER\_NUMBER NUMBER(10) NOT NULL,**

**CUSTOMER\_NAME VARCHAR2(20),**

**CUSTOMER\_TYPE VARCHAR2(20),**

**CONSTRAINT PK\_CUSTOMER PRIMARY KEY(CUSTOMER\_NUMBER),**

**CONSTRAINT CK\_CUSTOMER\_CTYPE CHECK (CUSTOMER\_TYPE IN('R','W','O')));**

**CREATE INDEX INDX\_CUSTOMER\_NAME ON PXGST128.CUSTOMER (CUSTOMER\_NAME);**

**2. Code an appropriate CREATE TABLE INVOICE, which illustrates the use on any Primary Key, Check or Foreign key Constraint, and an appropriate CREATE INDEX statement for Invoice Table could have been used to create the above INVOICE table. The invoice date (IDATE) should not be NULL and initialized to Today's date. You must prefix your table name by your schema name, or table owner name (7 points).**

**CREATE TABLE PXGST128.INVOICE**

**(INO NUMBER(10) NOT NULL,**

**EMP\_NO NUMBER(10) NOT NULL,**

**CUSTOMER\_NUMBER NUMBER(10),**

**IDATE DATE NOT NULL,**

**CONSTRAINT PK\_INVOICE\_NO PRIMARY KEY (INO),**

**CONSTRAINT FK\_EMPNO FOREIGN KEY (EMP\_NO) REFERENCES EMPLOYEE(EMP\_NO),**

**CONSTRAINT FK\_CNO FOREIGN KEY (CUSTOMER\_NUMBER) REFERENCES CUSTOMER (CUSTOMER\_NUMBER));**

**CREATE INDEX INDX\_IDATE ON PXGST128.INVOICE(IDATE);**

**3. Code three appropriate INSERT TABLE statements that could insert invoice number 1003 into the INVOICE table that would not violate any primary and foreign key constraints. You must prefix your table name by your schema name, or table owner name. (6 points).**

**INSERT INTO PXGST128.INVOICE (INO, EMP\_NO, CUSTOMER\_NUMBER, IDATE) VALUES ('1000', '99', '2', '05-JAN-2005' );**

**INSERT INTO PXGST128.INVOICE (INO, EMP\_NO, CUSTOMER\_NUMBER, IDATE) VALUES ('1001', '100', '2', '05-FEB-2005' );**

**INSERT INTO PXGST128.INVOICE (INO, EMP\_NO, CUSTOMER\_NUMBER, IDATE) VALUES ('1002', '99', '3', '05-FEB-2005' );**

**4. Code a SQL statement that would change the customer's name, i.e., CNAME of customer number 1 (PACKY) to PAT. You must prefix your table name by your schema name, or table owner name. (6 points).**

**UPDATE PXGST128.customer SET CUSTOMER\_NAME = 'PAT' WHERE CUSTOMER\_NAME = 'PACKY';**

**5. Code a SQL statement that would display the stock number and description of any inventory item that has a selling price greater than $5.00 or a Quantity of Hand > 50 units. You must prefix your table name by your schema name, or table owner name (6 points).**

**SELECT SNO,DESCR FROM PXGST128.INVENTORY WHERE QOH> 50;**

**6. Code a SQL statement that would display the invoice numbers, i.e., INO, for any invoice that have two or more line items. You must prefix your table name by your schema name, or table owner name (7 points).**

**SELECT COUNT(\*) NUM\_OF\_ITEMS,INO FROM**

**(SELECT \* FROM PXGST128.INVENTORY INNER JOIN ITEMS ON INVENTORY.SNO = ITEMS.SNO)**

**GROUP BY INO**

**HAVING COUNT(\*) = 2;**

**7. Code a SQL statement that would display Customer Number, Customer Name, Invoice Number, Invoice Date, Stock Number, Description, Quantity Shipped and Shipped Price for all Invoices in customer name order. You must prefix your table name by your schema name, or table owner name (10 points).**

**SELECT CUSTOMER.CUSTOMER\_NUMBER,CUSTOMER\_NAME,CUSTOMER\_TYPE,INVOICE.INO,INVOICE.IDATE,INVENTORY.SNO,INVENTORY.DESCR,ITEMS.QSHIP,SPRICE**

**FROM CUSTOMER**

**INNER JOIN INVOICE ON INVOICE.CUSTOMER\_NUMBER = CUSTOMER.CUSTOMER\_NUMBER**

**INNER JOIN INVENTORY ON INVENTORY.SNO = ITEMS.SNO**

**INNER JOIN ITEMS ON ITEMS.INO = INVOICE.INO;**

**8. Code a SQL statement that would display any stock number and description for any inventory item that was not sold. You must prefix your table name by your schema name, or table owner name. (5 points).**

SELECT SNO,DESCR FROM PXGST128.INVENTORY

WHERE SNO NOT IN(SELECT SNO FROM PXGST128.ITEMS);

**9. Code a SQL statement that will display the customer name and the total gross profit from sales of all the invoices. Hint: GROSS PROFIT = (SHIPPED PRICE - SHIPPED COST) \* QUANTITY SHIPPED. You must prefix your table name by your schema name, or table owner name. (7 points).**

**SELECT CUSTOMER.CUSTOMER\_NAME,**

**(ITEMS.SPRICE - ITEMS.SCOST) \* QSHIP AS GROSS\_PROFIT**

**FROM PXGST128.CUSTOMER**

**INNER JOIN PXGST128.INVOICE ON CUSTOMER.CUSTOMER\_NUMBER = INVOICE.CUSTOMER\_NUMBER**

**INNER JOIN PXGST128.ITEMS ON ITEMS.INO = INVOICE.INO;**

**10. Code a SQL statement that will display the stock number, description and the total gross profit from sales of all the invoices. Hint: GROSS PROFIT = (SHIPPED PRICE - SHIPPED COST) \* QUANTITY SHIPPED. You must prefix your table name by your schema name, or table owner name (7 points).**

SELECT INVENTORY.SNO,INVENTORY.DESCR,(ITEMS.SPRICE - ITEMS.SCOST) \* QSHIP AS GROSS\_PROFIT

FROM PXGST128.INVENTORY

INNER JOIN PXGST128.ITEMS ON ITEMS.SNO = INVENTORY.SNO;

**11. Code a SQL statement that will display the names of the table columns (not data) for the Invoice Table. You must prefix your table name by your schema name, or table owner name. (2 points)**

**DESC PXGST128.INVOICE;**

**12. Code n SQL statement that will display all of the indexes used in your database. You must prefix your table name by your schema name, or table owner name (4 points)**

**SELECT INDEX\_NAME FROM PXGST128.USER\_IND\_COLUMNS;**

**13. Display all of the constraints used in your database. You must prefix your table name by your schema name, or table owner name. (4 points)**

SELECT \* FROM PXGST128.USER\_CONSTRAINTS

**14. Code a SQL statement that will display CNO, CNAME, CTYPE, INO, EMPNO, and IDATE for all customers and invoices. You must prefix your table name by your schema name, or table owner name. (6 points)**

SELECT PXGST128.CUSTOMER.CUSTOMER\_NUMBER,CUSTOMER\_NAME,CUSTOMER\_TYPE,INO,EMP\_NO,IDATE

FROM PXGST128.CUSTOMER INNER JOIN PXGST128.INVOICE ON

CUSTOMER.CUSTOMER\_NUMBER = INVOICE.CUSTOMER\_NUMBER;

**15. Code a SQL statement that will display CNO, CNAME, CTYPE, INO, IDATE, EMPNO, ENAME, SNO, DESCR, QSHIP, SPRICE for all "wholesale" customers and invoices. You must prefix your table name by your schema name, or table owner name. (7 points)**

SELECT \* FROM

(SELECT CUSTOMER.CUSTOMER\_NUMBER,CUSTOMER\_NAME,CUSTOMER\_TYPE,INVOICE.INO,INVOICE.IDATE,EMPLOYEE.EMP\_NO,EMPLOYEE.EMPLOYEE\_NAME,

INVENTORY.SNO,INVENTORY.DESCR,ITEMS.QSHIP,ITEMS.SPRICE

FROM CUSTOMER

INNER JOIN INVOICE ON INVOICE.CUSTOMER\_NUMBER = CUSTOMER.CUSTOMER\_NUMBER

INNER JOIN EMPLOYEE ON EMPLOYEE.EMP\_NO = INVOICE.EMP\_NO

INNER JOIN INVENTORY ON INVENTORY.SNO = ITEMS.SNO

INNER JOIN ITEMS ON ITEMS.INO = INVOICE.INO)

WHERE CUSTOMER.CUSTOMER\_TYPE = 'W';