**CIS 310-75 TEST2 FALL 2010**

**NAME:\_\_\_SOLUTION\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. (30pts)Use the given diagram answer the following questions:
2. List all the partial dependencies if any. If you have found a partial dependency, use that partial dependency to explain why partial dependency is undesirable. You must provide sample data values to illustrate your point. NO credit otherwise.

Partial dependencies:

INV\_NUM->SALE\_DATE

PROD\_NUM->PROD\_DESCRIPTION, VEND\_CODE, VEND\_NAME, PROD\_PRICE

Partial dependencies are undesirable because they cause redundancies. For example in the second partial redundancy listed above the partial dependency would cause the same product data to be listed as many times as the product is ordered.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| INV\_NUM | PROD\_NUM | SALE\_DATE | PROD\_DESCRIPTION | VEND\_CODE | VEND\_NAME | PROD\_PRICE | NUM\_SOLD |
| 123 | ABC | 11/22/2010 | iPad 32 G Wifi | XYZ | Apple, Inc. | 599 | 1 |
| 456 | ABC | 11/24/2010 | iPad 32 G Wifi | XYZ | Apple, Inc. | 599 | 4 |

1. List all the transitive dependencies if any.

VEND\_CODE->VEND\_NAME

1. Is the given table in 1NF? Why?

YES, since every row is unique/the table has a primary key

1. Is the given table in 2NF? If not, normalize it into a 2NF database. Show ALL tables.

|  |  |
| --- | --- |
| INV\_NUM | SALE\_DATE |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PROD\_NUM | PROD\_DESCRIPTION | VEND\_CODE | VEND\_NAME | PROD\_PRICE |

|  |  |  |
| --- | --- | --- |
| INV\_NUM | PROD\_NUM | NUM\_SOLD |

1. Is the result in d in 3NF? If not, normalize it into a 3NF database. Show ALL tables.

|  |  |
| --- | --- |
| INV\_NUM | SALE\_DATE |

|  |  |  |  |
| --- | --- | --- | --- |
| PROD\_NUM | PROD\_DESCRIPTION | VEND\_CODE | PROD\_PRICE |

|  |  |  |
| --- | --- | --- |
| INV\_NUM | PROD\_NUM | NUM\_SOLD |

|  |  |
| --- | --- |
| VEND\_CODE | VEND\_NAME |

1. Draw a Crow’s Foot ERD for the resulting database in e.



The following questions are based on the given database. Please note the rows supplied in each table are only a subset of the data and you must use inner join (the new syntax) when appropriate. NO CREDIT OTHERWISE. These questions are worth 70 points.

1. List the name and available credit for every customer who has a credit limit of $1,500. Available credit is the difference between credit limit and balance.

SELECT CUSTOMER\_NAME, (CREDIT\_LIMIT – BALANCE) AS AVAILABLE\_CREDIT

FROM CUSTOMER

WHERE CREDIT\_LIMIT > 1500

1. List the name and address for every customer who lives on Pine street.

SELECT \*

FROM CUSTOMER

WHERE STREET LIKE ‘%PINE%’

1. Find the total number of customers and the total of their balances.

SELECT COUNT(\*), SUM(BALANCE)

FROM CUSTOMER

1. List the customer number of every customer who has an existing order. Do not repeat the same customer number.

SELECT DISTINCT CUSTOMER\_NUM

FROM ORDERS

1. List all customers whose credit limit is greater than the average credit limit.

SELECT \*

FROM CUSTOMER

WHERE CREDIT\_LIMIT > (SELECT AVG(CREDIT\_LIMIT) FROM CUSTOMER)

1. List the total value of each order. The value of each order line is quoted price \* number ordered.

SELECT O.ORDER\_NUM, (L.NUM\_ORDERED \* L.QUOTED\_PRICE)

FROM ORDERS O INNER JOIN ORDER\_LINE L ON O.ORDER\_NUM = L.ORDER\_NUM

GROUP BY O.ORDER\_NUM

1. List each credit limit and the number of customers with that limit if the number of customers with that limit is greater than 2.

SELECT CREDIT\_LIMIT, COUNT(\*)

FROM CUSTOMER

GROUP BY CREDIT\_LIMIT

HAVING COUNT(\*) > 2

1. Give a problem statement whose answer requires a full join. Your example must be meaningful in the context of the given database.

LIST ALL CUSTOMERS AND SALES REPS. INCLUDE CUSTOMERS THAT DON’T HAVE A SALES REP AND SALES REP THAT DO NOT HAVE CUSTOMERS.

SELECT \*

FROM CUSTOMER C FULL JOIN REP R ON C.REP\_NUM = R.REP\_NUM

1. Find the description of every part included in order number 12491 using a nested query.

SELECT DESCRIPTION

FROM PART

WHERE PART\_NUM IN (SELECT PART\_NUM FROM ORDER\_LINE WHERE ORDER\_NUM = 12491)

1. Find the order number and order date for every order that contains part number BT04 using the EXISTS operator.

SELECT O.ORDER\_NUM, O.ORDER\_DATE

FROM ORDERS O

WHERE EXISTS (SELECT \* FROM ORDER\_LINE L WHERE O.ORDER\_NUM = L.ORDER\_NUM AND

L.PART\_NUM = ‘BT04’)

1. List the customer name, order number, order date, and order total for every order with a total over $100.

SELECT C.CUSTOMER\_NAME, O.ORDER\_NUM, O.ORDER\_DATE,

SUM(L.NUM\_ORDERED \* L.QUOTED\_PRICE) AS TOTAL

FROM CUSTOMER C INNER JOIN ORDERS O ON C.CUSTOMER\_NUM = O.CUSTOMER\_NUM

INNER JOIN ORDER\_LINE L ON O.ORDER\_NUM = L.ORDER\_NUM

GROUP BY C.CUSTOMER\_NAME, O.ORDER\_NUM, O.ORDER\_DATE

HAVING SUM(L.NUM\_ORDERED \* L.QUOTED\_PRICE) > 100

1. Find every pair of customers who have the same first name and last name.

SELECT A.CUSTOMER\_NAME, B.CUSTOMER\_NAME

FROM CUSTOMER A INNER JOIN CUSTOMER B ON A.CUSTOMER\_NAME = B.CUSTOMER\_NAME

WHERE A.CUSTOMER\_NUM < B.CUSTOMER\_NUM

1. Use CREATE TABLE statements to define the Customer table and the Rep table.

CREATE TABLE [dbo].[REP](

[REP\_NUM] [varchar](2) NOT NULL,

[LAST\_NAME] [varchar](15) NULL,

[FIRST\_NAME] [varchar](15) NULL,

[STREET] [varchar](15) NULL,

[CITY] [varchar](15) NULL,

[STATE] [varchar](2) NULL,

[ZIP] [varchar](5) NULL,

[COMMISSION] [money] NULL,

[RATE] [float] NULL

)

CREATE TABLE [dbo].[CUSTOMER](

[CUSTOMER\_NUM] [varchar](3) NOT NULL,

[CUSTOMER\_NAME] [varchar](35) NULL,

[STREET] [varchar](15) NULL,

[CITY] [varchar](15) NULL,

[STATE] [varchar](2) NULL,

[ZIP] [varchar](5) NULL,

[BALANCE] [money] NULL,

[CREDIT\_LIMIT] [money] NULL,

[REP\_NUM] [varchar](2) NULL

)

1. Use ALTER TABLE statements to add the primary key and foreign key constraints to the two tables defined in the previous questions.

ALTER TABLE REP

ADD CONSTRAINT [PK\_REP] PRIMARY KEY (REP\_NUM)

ALTER TABLE CUSTOMER

ADD CONSTRAINT [PK\_CUSTOMER] PRIMARY KEY (CUSTOMER\_NUM),

CONSTRAINT [FK\_CUSTOMER\_REP] FOREIGN KEY([REP\_NUM])

REFERENCES [dbo].[REP] ([REP\_NUM])

1. Use an INSERT statement to add a row to the Customer table.

INSERT INTO CUSTOMER VALUES(‘123’, ‘A’, ‘B’, ‘C’, ‘D’, 12345, 100, 150, ‘03’)

1. Use an UPDATE statement to increase 10% the credit limit of all customers of sales rep William Smith (you don’t know his slsrep number).

UPDATE CUSTOMER

SET CREDIT\_LIMIT = CREDIT\_LIMIT \* 1.10

WHERE REP\_NUM IN (SELECT REP\_NUM FROM REP WHERE LAST\_NAME = ‘Smith’ AND

FIRST\_NAME = ‘William’)

1. List all parts (including part description) whether or not they have been ordered. If a part has

been ordered, then you must also list the total quantity ordered.

SELECT P.DESCRIPTION, SUM(ISNULL(L.NUM\_ORDERED, 0))

FROM PART P LEFT JOIN ORDER\_LINE L ON P.PART\_NUM = L.PART\_NUM

GROUP BY P.DESCRIPTION

Dependency diagram for Question 1

****The database for the SQL questions.

