**ISM 6212 Exam** **NAME** NICHOLAS CATANI

**1. (24 pts) (Enhanced) Entity Relationship Diagramming**

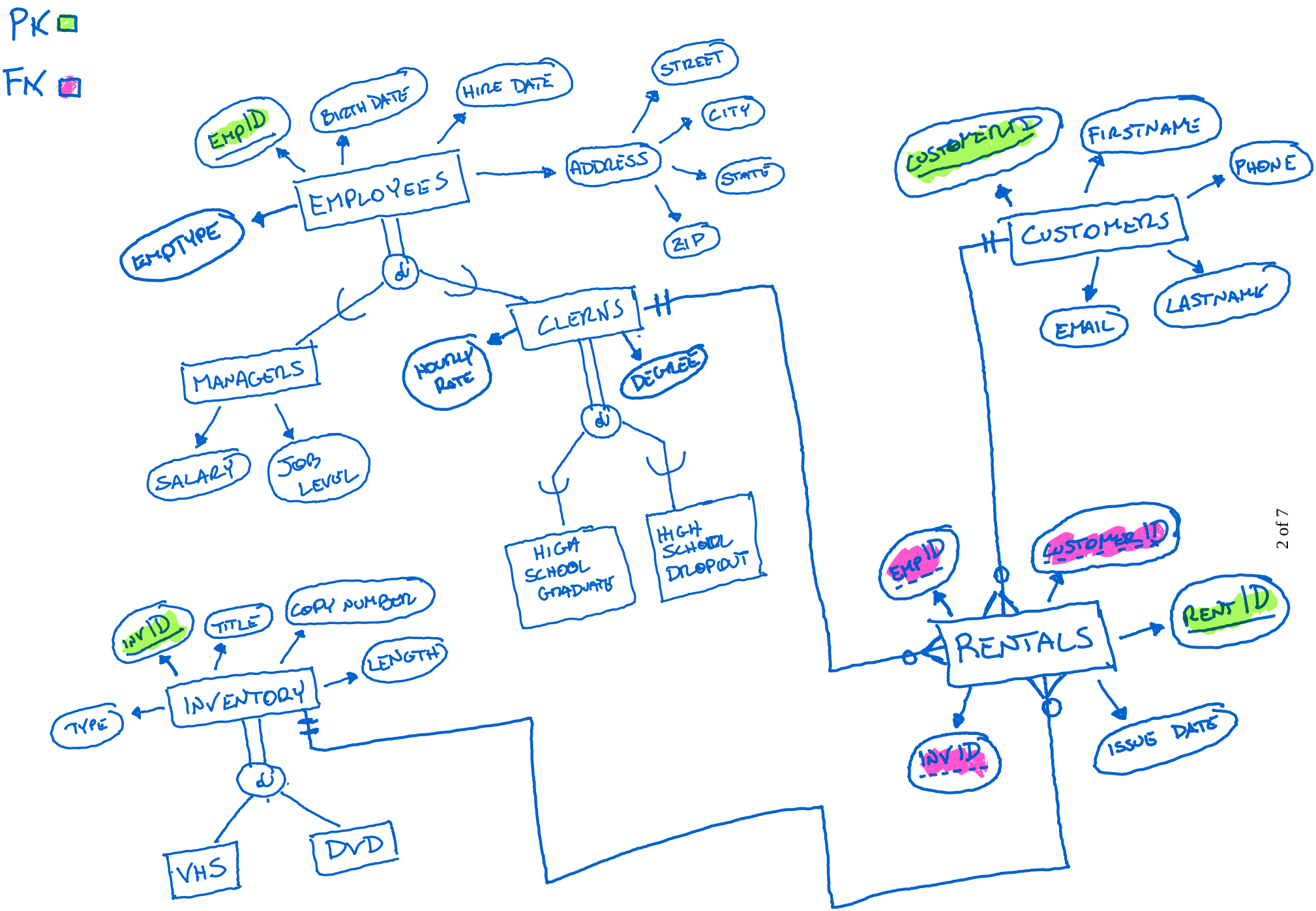
LVS, Inc. – a local video store – keeps information on employees, customers, inventory (videos), and rentals.

- There are two types of employees at the video store, managers and clerks. The following information is needed about all employees (EmpID, Birth Date, Address (Street, City, State, and Zip), and Hire Date). Additionally Managers are paid a yearly Salary and have a Job Level. Clerks have an Hourly Rate and the store would like to know if they have a High School Degree or not. An employee must be a manager or a clerk, not both. Only clerks can rent videos.
- The video store would like to keep the following information about its customers: A unique CustomerID, Name (First and Last), Phone, and Email. Assume that there are no dependent accounts.
- The Inventory of the video store is Video Cassettes (VHS) and Digital Video Discs (DVD). Inventory information needed is: a unique ID, Title, Copy Number, Length, and Type (vhs or dvd).
- Each rental needs to have a unique identifier and track the customer, the employee, the Date and *details* about which videos are rented.

Some business rules for LVS are:

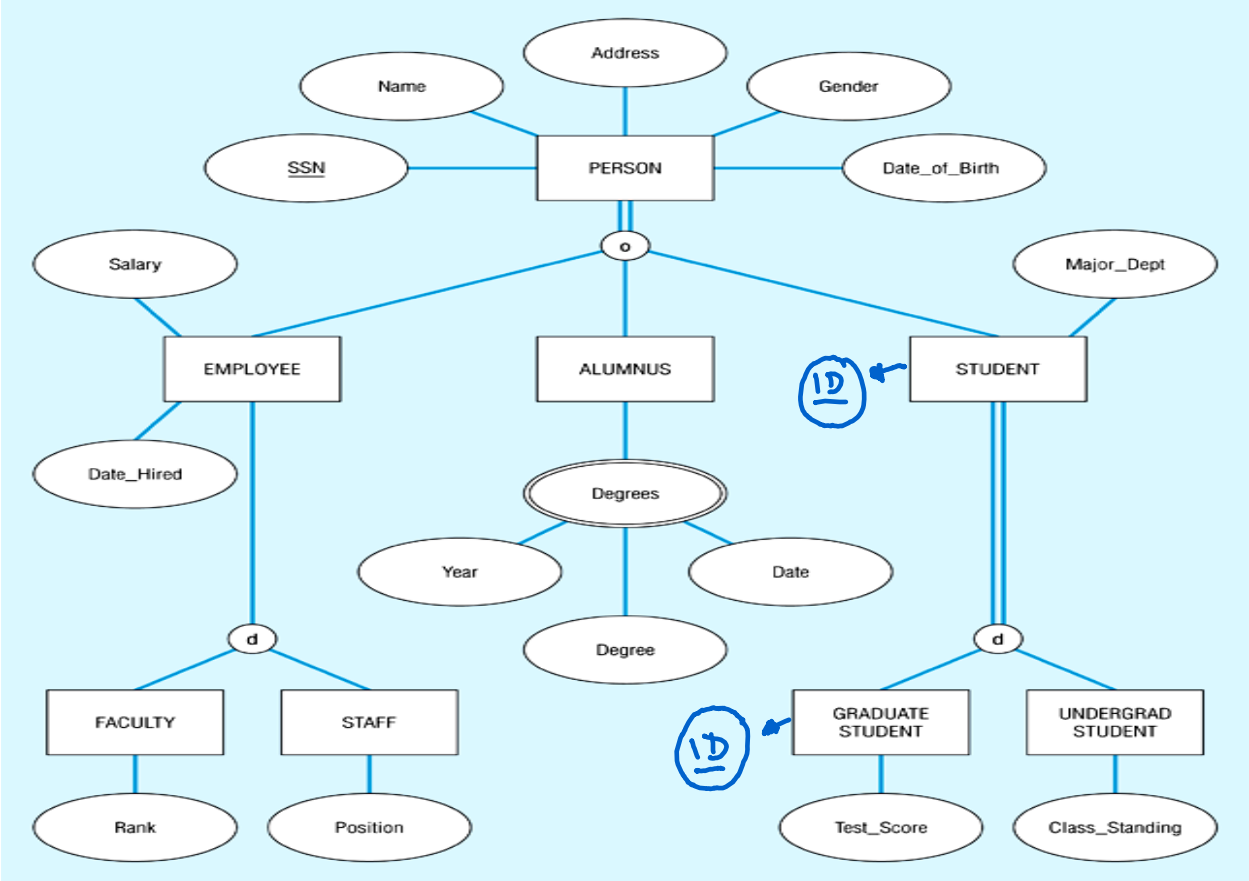
- It is important to know which employee rented what movie to what customer.
- For this exercise ignore the billing aspect.
- Please convert any Many-to-Many relationships you may have to associate entities.
- Please show all keys (primary and foreign).
- Please use the Chen ER modeling technique learned in class.

On the next page, draw an (E)ER diagram that is an appropriate data model for LVS.


PK
FK
EMPLOYEES
EMP ID
BIRTH DATE
HIRE DATE
EMPTYPE
ADDRESS
STREET
CITY
STATE
ZIP
d
MANAGERS
SALARY
JOB LEVEL
CLERKS
HOURLY RATE
DEGREE
HIGH SCHOOL GRADUATE
HIGH SCHOOL DROPOUT
CUSTOMERS
CUSTOMER ID
FIRSTNAME
PHONE
EMAIL
LASTNAME
RENTALS
EMP ID
CUSTOMER ID
RENT ID
INV ID
ISSUE DATE
INVENTORY
INV ID
TITLE
COPY NUMBER
LENGTH
TYPE
VHS
DVD

**2. (6 pts) Enhanced Entity Relationship Diagramming**

Given the following (incomplete) EER diagram answer the following questions:




A) Can I be both a graduate student and an undergraduate student?__NO__

B) Can I be both a faculty and a staff employee?__NO__

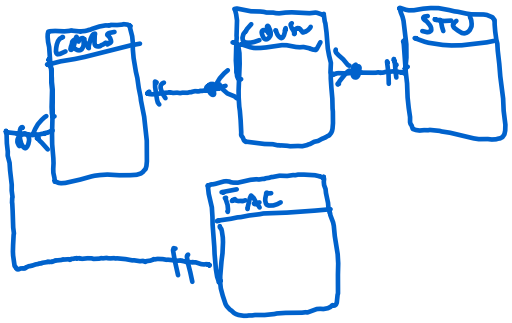
C) Can I be both an employee and an undergraduate student?__YES__

D) Can I be an employee other than a faculty or a staff?__YES__

E) How many (final total) attributes (shown and not shown) does a graduate student have?__13__

F) Name all of the attributes (shown and not shown) of a faculty member.

SSN, NAME, ADDRESS, GENDER, DATE OF BIRTH, RANK, DATE_HIRED, SALARY, FACULTY ID, ALUMNUS, STUDENT




**3. SQL (30 pts)**

The relational schema for a university database is:

STUDENTS (stuID, stuName, Major, Credits)
COURSES (crsID, facID, Schedule, Room)
FACULTY (facID, facName, Dept, Rank)
ENROLL (enrID, crsID, stuID, Grade)

Answer the following questions using SQL, pay particular attention to the table names and field names, points will be deducted if the exact names are not used.

(a) Find a count of all the students by major. The results should be a two-column table with Major and Number_of_Students. (7 pts)

```
SELECT MAJOR, COUNT(STUID) AS [NUMBER OF STUDENTS]
FROM STUDENTS
GROUP BY MAJOR ;
```

(b) Find all of the students, the courses they took and the grade they received, sorted first by student then by course. The results should be a three-column table with stuName, crsID, and Grade. (7 pts)

```
SELECT STUID, CRSID, GRADE
FROM ENROLL
ORDER BY STUID, CRSID ;
```

(c) Find an alphabetical list of all the students who have taken a class taught by Adams (assume there is only one faculty member named Adams). The results should be a single column table with the label Students_of_Adams. (8 pts)

{4 TABLE
3 ASS.

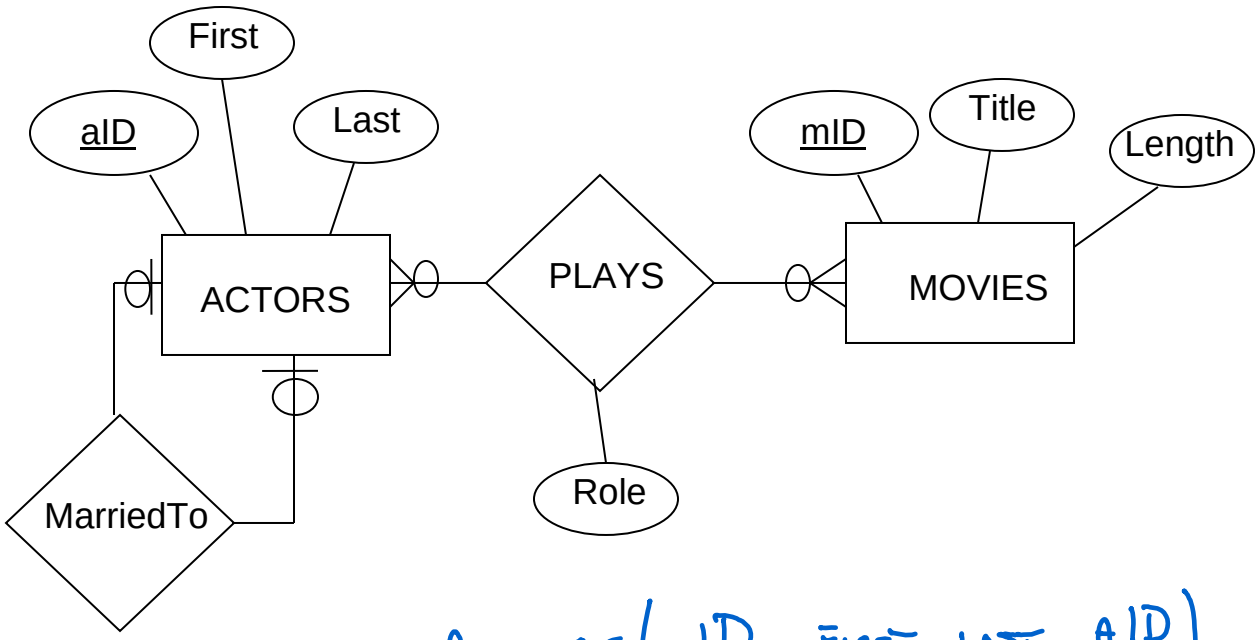
```
SELECT S.STUID AS [STUDENTS OF ADAMS]
  FROM FACULTY F, COURSES C, ENROLL E, STUDENTS S
WHERE F.FACID = C.FACID AND C.CRSID = E.CRSID
 AND E.STUID = S.STUID
ORDER BY S.STUID ;
```

(d) Write a SQL statement that will add the following data into the Faculty table: (id=123, name=LaBrie, rank=Associate, and department=Business). (8 pts)

```
INSERT INTO FACULTY (FACID, FACNAME, DEPT, RANK)
VALUES ('123', 'LABRIE', 'BUSINESS', 'ASSOCIATE') ;
```

**4. Converting ERDs to Schema (10 pts)**

Given the following ERD, write out the relational schema. Assume an actor can play more than one role in a movie.




ACTORS(AID, FIRST, LAST, AID)

PLAYS(AID, MID, ROLE)

MOVIES(MID, TITLE, LENGTH)

**4.5 Querying an ERD (8 pts)**

Write a SQL Statement that gives me a count of all the actors that have played the same role in more than one movie (for example Emma Watson played Hermione Granger 8 times). The output should have four columns: First, Last, Role, and CountOfRole.

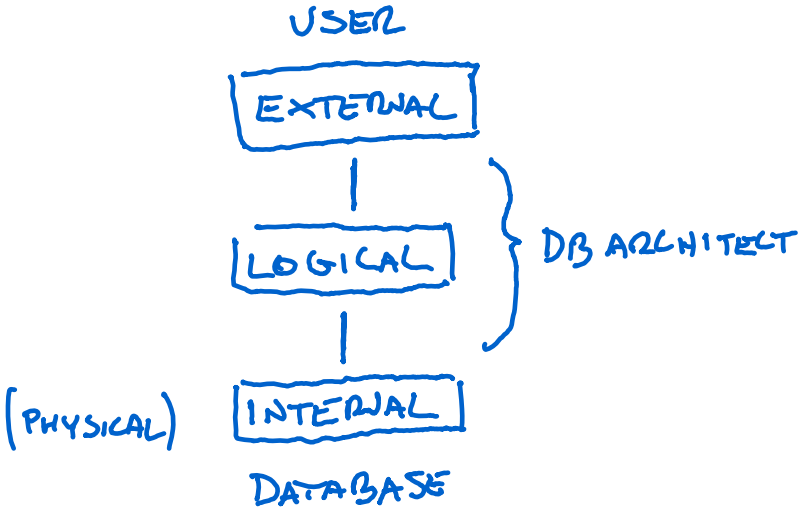
{ 3 TAB
2 ASS

```
SELECT FIRST, LAST, ROLE, COUNT(TITLE) AS COUNTOFROLE
FROM ACTORS AS A, PLAYS AS P, MOVIES AS M
WHERE A.AID = P.AID AND M.MID = P.MID
GROUP BY FIRST, LAST, ROLE;
```

**5. Three Schema Architecture (6 pts)**

In your own words describe what the 3-Schema Architecture is, what some of its benefits are, and what are some of the roles that people play in it.

IT IS A FRAMEWORK FOR MANAGING ACCESS TO DATA THAT INVOLVES THREE SCHEMAS.




- IT ALLOWS EACH USER TO ACCESS THE SAME DATABASE WITH A DIFFERENT CUSTOMIZED VIEW OF DATA
- IT ALLOWS AN ADMIN THE DATABASE STORAGE STRUCTURE WITHOUT CAUSING DOWNTIME FOR OTHER USERS.
- PROTECTION FROM DATA LOSS BY ACCESSING THE OTHER SCHEMA

**6. Normalization (6 pts)**

What is the purpose of normalization? Describe the process, what objects do we perform it on, and explain the first three normal forms.

BASICALLY, IT IS A DESIGN TECHNIQUE THAT REDUCES DATA REDUNDANCY AND INCREASES DATA INTEGRITY. IT DIVIDES LARGER TABLES INTO SMALLER ONES AND LINKS THEM USING RELATIONSHIPS.

1NF) EACH RECORD NEEDS TO BE UNIQUE AND NO MORE THAN A SINGLE VALUE ASSOCIATED WITH EACH TABLE CELL.

2NF) INTRODUCES REFERTIAL INTEGRITY (INSERT VALUES INTO FOREIGN KEY THAT EXIST IN THE PRIMARY KEY IN THE PARENT TABLE) AND TRANSITIVE FUNCTIONAL DEPENDENCY.

3NF) THERE'S NO TRANSITIVE FUNCTIONAL DEPENDENCIES (CHANGING A VALUE IN A NON-KEY COLUMN, WON'T CAUSE CHANGES IN THE OTHER NON-KEY COLUMNS)

**7. JOINS (10 pts)**

Assume that **CATEGORIES** and **PRODUCTS** tables contain the following information:

**CATEGORIES**

| catID | catNAME |
|---|---|
| c1 | Beverages |
| c2 | Dinners |
| c3 | Desserts |

**PRODUCTS**

| prodID | prodName | prodPrice | catID |
|---|---|---|---|
| p1 | Pizza | 5.00 | c2 |
| p2 | Root Beer | 3.50 | c1 |
| p3 | Ice Cream | 2.99 | c3 |
| p4 | Burritos | 2.00 | c2 |

3 x 4 = 12

(a) In the space below, show the output (WITH ALL OF ITS DATA) that would be the result of the following query.

```
SELECT *
FROM CATEGORIES, PRODUCTS;
```

| CAT.CATID | CATNAME | PROD.CATID | PRODID | PROD.NAME | PROD.PRICE |
|---|---|---|---|---|---|
| C1 | BEVERAGES | C2 | P1 | PIZZA | 5.00 |
| C2 | DINNERS | C2 | P1 | PIZZA | 5.00 |
| C3 | DESSERTS | C2 | P1 | PIZZA | 5.00 |
| C1 | BEVERAGES | C1 | P2 | ROOT BEER | 3.50 |
| C2 | DINNERS | C1 | P2 | ROOT BEER | 3.50 |
| C3 | DESSERTS | C1 | P2 | ROOT BEER | 3.50 |
| C1 | BEVERAGES | C3 | P3 | ICE CREAM | 2.99 |
| C2 | DINNERS | C3 | P3 | ICE CREAM | 2.99 |
| C3 | DESSERTS | C3 | P3 | ICE CREAM | 2.99 |
| C1 | BEVERAGES | C2 | P4 | BURRITOS | 2.00 |
| C2 | DINNERS | C2 | P4 | BURRITOS | 2.00 |
| C3 | DESSERTS | C2 | P4 | BURRITOS | 2.00 |

(b) In the space below, show the output (WITH ALL OF ITS DATA) that would be the result of the following query.

```
SELECT *
FROM CATEGORIES, PRODUCTS
WHERE CATEGORIES.catID = PRODUCTS.catID;
```

| CAT.CATID | CATNAME | PROD.CATID | PRODID | PRODNAME | PROD.PRICE |
|---|---|---|---|---|---|
| C1 | BEVERAGES | C1 | P2 | ROOT BEER | 3.50 |
| C2 | DINNERS | C2 | P1 | PIZZA | 5.00 |
| C2 | DINNERS | C2 | P4 | BURRITOS | 2.00 |
| C3 | DESSERTS | C3 | P3 | ICE CREAM | 2.99 |

NOTE: This exam has 100 points, your score will be multiplied by 2.5 for a maximum of 250 points for the course.