

B. PROBLEMS.

1. Given the followi  
Empl  
C

A. MULTIPLE CHOICE.

Circle the choice that best answer each question. Grading: Correct = 2 points, Incorrect=0 points,

no answer = 0

1. SQL does not include
  - A) A query language
  - B) A schema definition language
  - C) A programming language
  - D) A data manipulation language
2. SELECT R.a,R.b from R join S using c assumes that
  - A) c is a field of R but not of S
  - B) c is a field of R and S
  - C) c is a field of S but not R
  - D) c is not a command field
3. Which of the following SQL instructions might have duplicates
  - A) UNION
  - B) INTERSECT
  - C) JOIN
  - D) EXCEPT
4. Select a,b from R union Select c,d from S produces a table with
  - A) two columns
  - B) three columns
  - C) no columns
  - D) four columns
5. A condition on count(\*) can be included in a SQL query after GROUP BY using
  - A) HAVING
  - B) WHERE
  - C) CASE
  - D) IF
6. Result tables from SQL queries
  - A) can have duplicates
  - B) cannot have duplicates
  - C) are always sorted by id
  - D) always have a key
7. A foreign key must reference
  - A) any field combination of another table
  - B) all the primary key fields of another table
  - C) some of the primary key fields of another table
  - D) just one field of another table, even if it is not the complete primary key
8. SELECT \* FROM A,B; computes
  - A)  $A \cup B$
  - B)  $A - B$
  - C)  $A \times B$
  - D)  $A \cap B$
9. Result table in SQL are
  - A) Sets
  - B) Relations
  - C) Lists
  - D) Queries
10. SQL stands for
  - A) Sequel
  - B) Structured Query Language
  - C) Relational Database System
  - D) Simple Query Logic

(10)

## B. PROBLEMS.

1. Given the following tables:

- Employees(ssn, name, address, isFaculty)
- Contact(phone, ssn)
- University(uname, campus, uphone, uaddress)
- StudyAt(ssn, uname, campus, major, minor)
- Documents(docid, issueDate, type, ssn, uname, campus)

Assume that type can be 'diploma', 'certificate', or 'transcript'.

(15)

Write complete SQL instructions to create the Database Schema. If you need assumptions, please write them down.

- Create table Employees (ssn int, name varchar(20), address varchar(20), isFaculty bool, primary key (ssn));
- create table Contact (ssn int, phone varchar(20), foreign key (ssn) references Employees(ssn), primary key (ssn, phone));
- create table University (uname varchar(20), campus varchar(20), uphone varchar(20), uaddress varchar(20), primary key (uname, campus))
- create table StudyAt (ssn int, uname varchar(20), campus varchar(20), major varchar(20), minor varchar(20), foreign key (ssn) references Employees(ssn), foreign key (uname, campus) references University(uname, campus), primary key (ssn, uname, campus));
- create table Documents (docid int, issueDate date, type enum ('diploma', 'certificate', 'transcript'), ssn int, uname varchar(20), campus varchar(20), foreign key (ssn) references Employees(ssn), foreign key (uname, campus) references University(uname, campus), primary key (docid));

2. Given the table:

- products(storeid, productid, productType, price, quantity)

productType can be 'computer', 'appliance', or 'accessory'

Provide SQL instructions that produce a table where the columns are product types, the rows are stores (storeId). The content of each entry is the total number (quantity) of products of that particular type in a given store.

Select storeid,  
sum(if(productType = 'computer', quantity, 0)) computer - 9,  
sum(if(productType = 'appliance', quantity, 0)) appliance - 9,  
sum(if(productType = 'accessory', quantity, 0)) accessory - 9  
from products group by store id;

(20)

3. Suppose relations  $R$  and  $S$  have  $n$  tuples and  $m$  tuples, respectively. Give the minimum and maximum number of tuples that the results of the following expressions can have.

- (a)  $R \cup S$
- (b)  $R \bowtie S$
- (c)  $\sigma_C(R) \times S$ , for some condition  $C$ .
- (d)  $\pi_L(R) - S$ , for some list of attributes  $L$ .

$R = n$  tuples

$S = m$  tuples

	min	max
a)	$\max(n, m)$	$n+m$
b)	0	$n+m$
c)	0	$n+m$
d)	0	$n$

15

(1)

4. Given the following database:

$\text{Product}(\text{maker}, \text{model}, \text{type})$   
 $\text{PC}(\text{model}, \text{speed}, \text{ram}, \text{hd}, \text{cd}, \text{price})$   
 $\text{Laptop}(\text{model}, \text{speed}, \text{ram}, \text{hd}, \text{screen}, \text{price})$   
 $\text{Printer}(\text{model}, \text{color}, \text{type}, \text{price})$

write the following queries using relational algebra. You can use intermediate relations in the case of harder expressions. **IMPORTANT:** Model numbers are not unique to the product. There could a PC with the same model number as a Laptop.

(a) Find the model number and price of all products (of any type) made by a given manufacturer "IBM".

~~$\Pi_{\text{model}, \text{price}} (\text{Product} \bowtie \text{PC}) \cup \Pi_{\text{model}, \text{price}} (\text{Product} \bowtie \text{Laptop}) \cup \Pi_{\text{model}, \text{price}} (\text{Product} \bowtie \text{Printer})$~~

$\Pi_{\text{model}, \text{price}} (\Pi_{\text{model}, \text{price}} (\delta_{\text{maker} = 'IBM'} (\text{Product} \bowtie \text{PC})) \cup \Pi_{\text{model}, \text{price}} (\delta_{\text{maker} = 'IBM'} (\text{Product} \bowtie \text{Laptop}))$   
 $\cup \Pi_{\text{model}, \text{price}} (\delta_{\text{maker} = 'IBM'} (\text{Product} \bowtie \text{Printer})))$

(b) Find the model number of all color laser printers.

$\Pi_{\text{model}} (\delta_{\text{color} = 'true'} (\text{Printer}))$

(2)

(c) Find the manufacturer, model and price of all PCs with a 6X or a 8X CD.

$\Pi_{\text{maker}, \text{model}, \text{price}} (\delta_{\text{cd} = '6X'} (\text{Product} \bowtie \text{PC}) \cup \delta_{\text{cd} = '8X'} (\text{Product} \bowtie \text{PC}))$

(d) Find those manufacturers that sell Laptops but not PCs.

$\Pi_{\text{maker}} (\text{Product} \bowtie \text{Laptop}) - \Pi_{\text{maker}} (\text{Product} \bowtie \text{PC})$

5. Repeat the queries of Problem 4 using SQL.

① select f.model, f.price from

(( select a.model, a.price from (Product join PC using(model)) as a  
where maker = 'IBM')

UNION

( select b.model, b.price from (Product join Laptop using(model)) as b  
where maker = 'IBM')

UNION

( select c.model, c.price from (Product join Printer using(model)) as c  
where maker = 'IBM')) as f;

② Select model from Printer where color = 'true';

③ Select f.maker, f.model, f.price from

(( select a.maker, b.model, c.price from (Product join PC using(model))  
as a where a.CD = '6X')

UNION

( select b.maker, b.model, b.price from (Product join PC using(model))  
as b where b.CD = '8X')) as f;

④ Select f.maker from (Product join Laptop using(model)) as f

where f.maker not in

( select a.maker from (Product join PC using(model)) as a);