CMPUT 291
Midterm Examination – Oct 13, 2016

Section: A1

Instructor: Joerg Sander

Name _____

Signature _____

Instructions:

1. Turn off your cell phone.
2. Sign your name on the signature line.
3. Place your student id number in the designated spot on the top of the **next** page.
4. The time for this test is 1 hour.
5. The exam is "closed-book", i.e., no references, notes, books, calculators, or smart phones are allowed.
6. Place all answers in this booklet and do not hand in any other work.

Student Id Number: _____

Marks obtained:

Part 1	Part 2	Part 3	Total
/12	/18	/30	/60

Part 1 [12 marks]

1. [4 marks] List 4 important advantages of a DBMS in comparison to using flat files to store and manage data. No detailed explanations are needed, just a keyword or short phrase.

1. **Data Independence**
2. **Concurrent Access to the data**
3. **Crash Recovery**
4. **Reduced Application development time, High Level Query Language, Query Optimization.**

2. [2 marks] What does "physical data independence" refer to in the context of DBMSs? Select only one of the following statements.

- ☐ Changes in the physical schema of several tables can be made independently of each other.
- ☒ The physical schema can be changed without affecting applications.
- ☐ The external schema can be changed without affecting the physical schema.
- ☐ The conceptual schema can be changed without affecting the applications.

3. [3 marks] Consider the following statements, assuming a relation schema R=(A, B, C, D):

1. If A is a candidate key of R, then (A,B) uniquely identifies a row in a table.
2. If A is a candidate key of R, then (A,B) is not a candidate key of R.
3. If (A,B) is a primary key of R, then A is a candidate key of R.
4. (A,B) cannot be a primary key of R because primary keys must consist of a single attribute.

Which of the following assertions is correct? Select only one.

- ☐ Statements 1, 3, and 4 are correct, and statement 2 is incorrect.
- ☐ Statements 1 and 3 are correct, and statements 2 and 4 are incorrect.
- ☒ Statements 1 and 2 are correct, and statements 3 and 4 are incorrect.
- ☐ Statements 1, 2, and 4 are correct, and statement 3 is incorrect.
- ☐ All are correct.
- ☐ All are incorrect.

4. [3 marks] Consider the following statements about foreign key constraints.

1. A table can have several foreign key constraints.
2. If a foreign key consists of more than one attribute, then each of these individual attributes is a candidate key in the referenced table.
3. A field with a foreign key constraint can contain NULL values.
4. The foreign key of a table must be the primary key of the referenced table.

Which of the following assertions is correct? Select only one.

- ☒ Statements 1, 3, and 4 are correct, and statement 2 is incorrect.
- ☐ Statements 1 and 4 are correct, and statements 2 and 3 are incorrect.
- ☐ Statements 1 and 2 are correct, and statements 3 and 4 are incorrect.
- ☐ Statements 1, 2, and 4 are correct, and statement 3 is incorrect.
- ☐ All are correct.
- ☐ All are incorrect.

Part 2 [18 marks]

1. [10 marks] Complete the ER diagram below according to the following specifications. Be sure to underline the primary keys in your diagram, and indicate any participation and key roles.
The ER diagram represents a simplified model of an insurance company: Each insurance policy is characterized by an ID and a description, and there are two kinds of insurance policies: a car insurance, which has also a rate, and a life insurance, which has also an amount and a time period. The company has customers, characterized by an SIN, a name, and an address. Each customer can have zero or one life insurance policy. If the customer has life insurance, the company wants to also store the date when the contract for this insurance was signed.
The company stores information about cars that have a model name, a (unique) vehicle identification number (VIN), and the fabrication year. The company does not want to store any information about cars which are not insured by one of their customers.
A car insurance policy covers the insurance of one car for a given customer. A customer can insure more than one car, but each car in the database is covered by exactly one car insurance policy. The date when the contract for a car insurance policy was signed should be recorded.

2 Primary keys

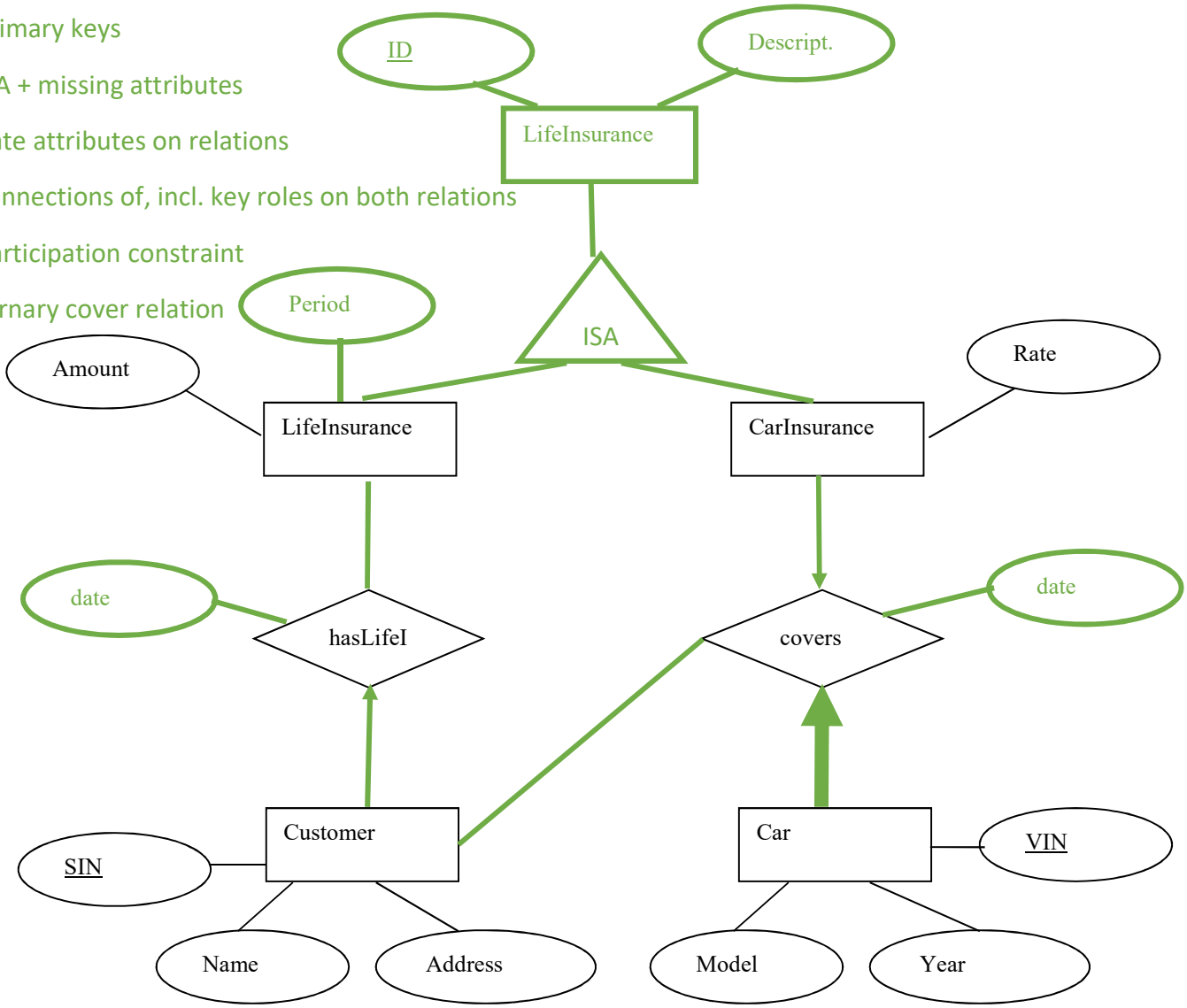
2 ISA + missing attributes

1 date attributes on relations

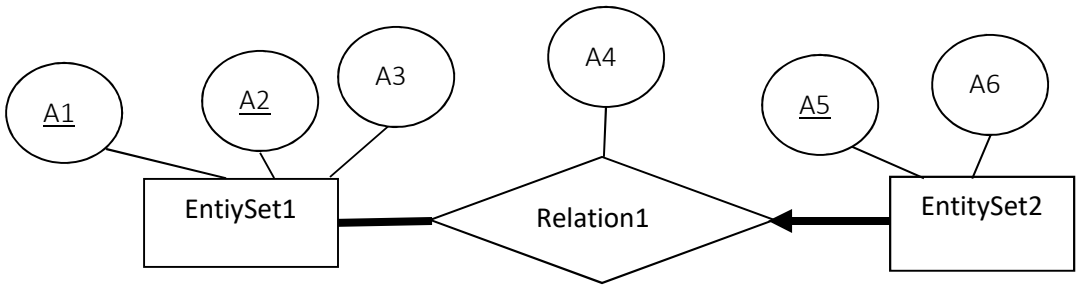
2 connections of, incl. key roles on both relations

1 participation constraint

2 ternary cover relation



2. [8 marks] Translate the following ER diagram into relational tables using CREATE TABLE commands, so that as much of the information in the diagram as possible is represented (and nothing else that is not expressed in the diagram).




```
CREATE TABLE E1 (
   A1 INT,
   A2 INT,
   A3 INT,
   PRIMARY KEY (A1, A2));

CREATE TABLE E2_R (
   A1 INT NOT NULL,
   A2 INT NOT NULL,
   A4 INT,
   A5 INT,
   A6 INT,
   PRIMARY KEY (A5),
   FOREIGN KEY (A1, A2) REFERENCES E1;
```

Part 3 [30 marks]

For the queries in this part, consider the following relational tables about suppliers, parts, and prices (the prices table contains information about which supplier can supply which part for a certain price).

Supplier

SupplierID	SName
170	Fraser Inc.
175	Heavy Screw
180	Kidd & Son
185	Schuster
186	Blue Steel
…	…

Parts

PartID	PName
001	IGZ-3
002	IGZ-7
003	IZZ-3
004	GZI-3
…	…

Prices

SupplierID	PartID	Price
170	001	12
170	003	20
185	001	11
185	002	19
185	003	22
…	…	…

1. [4 marks] Complete the following ***SQL*** query so that it finds the SupplierID of suppliers who do not supply the part with PartID 001.

```
SELECT  SupplierID FROM Supplier
EXCEPT
SELECT  SupplierID FROM Prices WHERE PartID = "001 "
```

2. [4 marks] Formulate the same query as in (a) (i.e., find the SupplierID of suppliers who do not supply the part with PartID 001) as a ***Relational Algebra*** query, using the minus operator (as shown), by completing the following expression.

$\pi_{\text{SupplierId}}(\text{Supplier}) - \pi_{\text{SupplierId}}(\sigma_{\text{PartID='001'}}(\text{Prices}))$

Supplier

SupplierID	SName
170	Fraser Inc.
175	Heavy Screw
180	Kidd & Son
185	Schuster
186	Blue Steel
…	…

Parts

PartID	PName
001	IGZ-3
002	IGZ-7
003	IZZ-3
004	GZI-3
…	…

Prices

SupplierID	PartID	Price
170	001	12
170	003	20
185	001	11
185	002	19
185	003	22
…	…	…

3. [4 marks] Complete the following ***SQL*** query so that it lists the names and the prices of all parts that are supplied by the supplier "Fraser Inc."

```
SELECT Pa.PName, Pr.Price
FROM   Prices Pr, Parts Pa, Supplier S
WHERE  Pa.PartID = Pr.PartID
       AND Pr.SupplierID = S.SupplierID
       AND S.SName = 'Fraser Inc.'
```

Alternative Solution

```
…FROM Prices Pr, Parts Pa WHERE Pa.PartID = Pr.PartID AND SupplierID IN
                                (SELECT SupplierID FROM Supplier WHERE SName = 'Fraser Inc.')
```

4. [4 marks] Formulate the same query as in (a) (i.e., list the names and the prices of all parts that are supplied by the supplier "Fraser Inc.") as a ***Relational Algebra*** query, ***using natural joins*** **(!)**, by completing the following expression.

$\pi_{\text{PName, Price}}$ ($\sigma_{\text{SName='Fraser Inc'}}$ (Prices $\bowtie$ (Parts $\bowtie$ Supplier)))

Supplier

SupplierID	SName
170	Fraser Inc.
175	Heavy Screw
180	Kidd & Son
185	Schuster
186	Blue Steel
…	…

Parts

PartID	PName
001	IGZ-3
002	IGZ-7
003	IZZ-3
004	GZI-3
…	…

Prices

SupplierID	PartID	Price
170	001	12
170	003	20
185	001	11
185	002	19
185	003	22
…	…	…

5. [3 marks] Complete the following ***SQL*** query by filling into each open spot [ ] the letter corresponding to the appropriate line from the list of choices, so that the query lists for each supplier the SupplierID and the number of different parts which are supplied by that supplier; the list should ordered by SupplierID (in ascending order).

```
SELECT   B
FROM Prices
D
I
```

Choices:

```
A. SupplierID, PartID, COUNT(*)
B. SupplierID, Count(PartID)
C. SupplierID, SUM(PartID)
D. GROUP BY SupplierID
E. GROUP BY PartID
F. GROUP BY SupplierID, PartID
G. ORDER BY PartID, SupplierID
H. ORDER BY SupplierID, PartID
I. ORDER BY SupplierID
```

Supplier

SupplierID	SName
170	Fraser Inc.
175	Heavy Screw
180	Kidd & Son
185	Schuster
186	Blue Steel
…	…

Parts

PartID	PName
001	IGZ-3
002	IGZ-7
003	IZZ-3
004	GZI-3
…	…

Prices

SupplierID	PartID	Price
170	001	12
170	003	20
185	001	11
185	002	19
185	003	22
…	…	…

6. [3 marks] Complete the following ***SQL*** query so that it lists all pairs of SupplierIDs and PartIDs for which the supplier supplies the part for the *minimum Price* among all suppliers for that part.

```
SELECT P.SupplierID, P.PartID
FROM Prices P
WHERE P.Price = (SELECT MIN(P1.Price) FROM Prices P1 WHERE P1.PartID=P.PartID)
```

7. [2 marks] Consider the following SQL query

```
SELECT DISTINCT P1.PartId
FROM Prices P1
WHERE NOT EXISTS
          (SELECT * from Prices P2 WHERE P1.Price < P2.Price)
```

What is the result of this query? Select only one option.

- ☐ The empty set, i.e., *no* tuples satisfy this condition.
- ☐ The list of *all* distinct PartIDs in the Prices table, i.e. the WHERE condition is always True.
- ☒ The PartIDs of the parts with the highest Price in the Prices table.
- ☐ The PartIDs of the parts with the lowest Price in the Prices table.
- ☐ The PartIDs of the parts that have at least two different Prices in the Prices table.

Supplier

SupplierID	SName
170	Fraser Inc.
175	Heavy Screw
180	Kidd & Son
185	Schuster
186	Blue Steel
…	…

Parts

PartID	PName
001	IGZ-3
002	IGZ-7
003	IZZ-3
004	GZI-3
…	…

Prices

SupplierID	PartID	Price
170	001	12
170	003	20
185	001	11
185	002	19
185	003	22
…	…	…

8. [6 marks] Write an ***SQL*** query that lists the PartID, the minimum Price, the maximum Price, and the average Price for each part which has more than 1 supplier and whose part name starts with "GZI".

```
SELECT Pr.PartID, MIN(Pr.Price), MAX(Pr.Price), AVG(Pr.Price)
FROM Prices Pr, Parts Pa
WHERE Pr.PartID = Pa.PartID AND Pa.PName LIKE "GZI%"
GROUP BY Pr.PartID
HAVING COUNT(*) > 1
```