CHALMERS
EXAMINATION / TENTAMEN

Course code/kurskod	Course name/kursnamn			
DAT335	DATA MANAGEMENT			
Anonymous code Anonym kod		Examination date Tentamensdatum	Number of pages Antal blad	Grade Betyg
DAT335-001-WTX		20190318	8	5

* I confirm that I've no mobile or other similar electronic equipment available during the examination.
Jag intygar att jag inte har mobiltelefon eller annan liknande elektronisk utrustning tillgänglig under eximinationen.

Solved task Behandlade uppgifter No/nr		Points per task Poäng på uppgiften	Observe: Areas with bold contour are to completed by the teacher. Anmärkning: Rutor inom bred kontur ifylles av lärare.
1	X	10	
2	X	7	
3	X	11	
4	X	18	
5	X	~~25~~	
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
Bonus poäng			
Total examination points Summa poäng på tentamen		90 ~~89~~	

2

1) Meta-data is information accompanying the data to somewhat describe what it means. May be labels among others.

3 An example would be column names, since it is not real data (only the rows are data), but describes what the data means. + more examples? XML?

2) BEGIN: Begin a transaction that is not actually performed until a commit

COMMIT: Apply the pending transaction to the database ✓

4 ROLLBACK: Cancel the current transaction, not performing changes to the database

3) A prepared statement gets precompiled before sent to th DBMS, rather than executing a query on the DBMS directly.

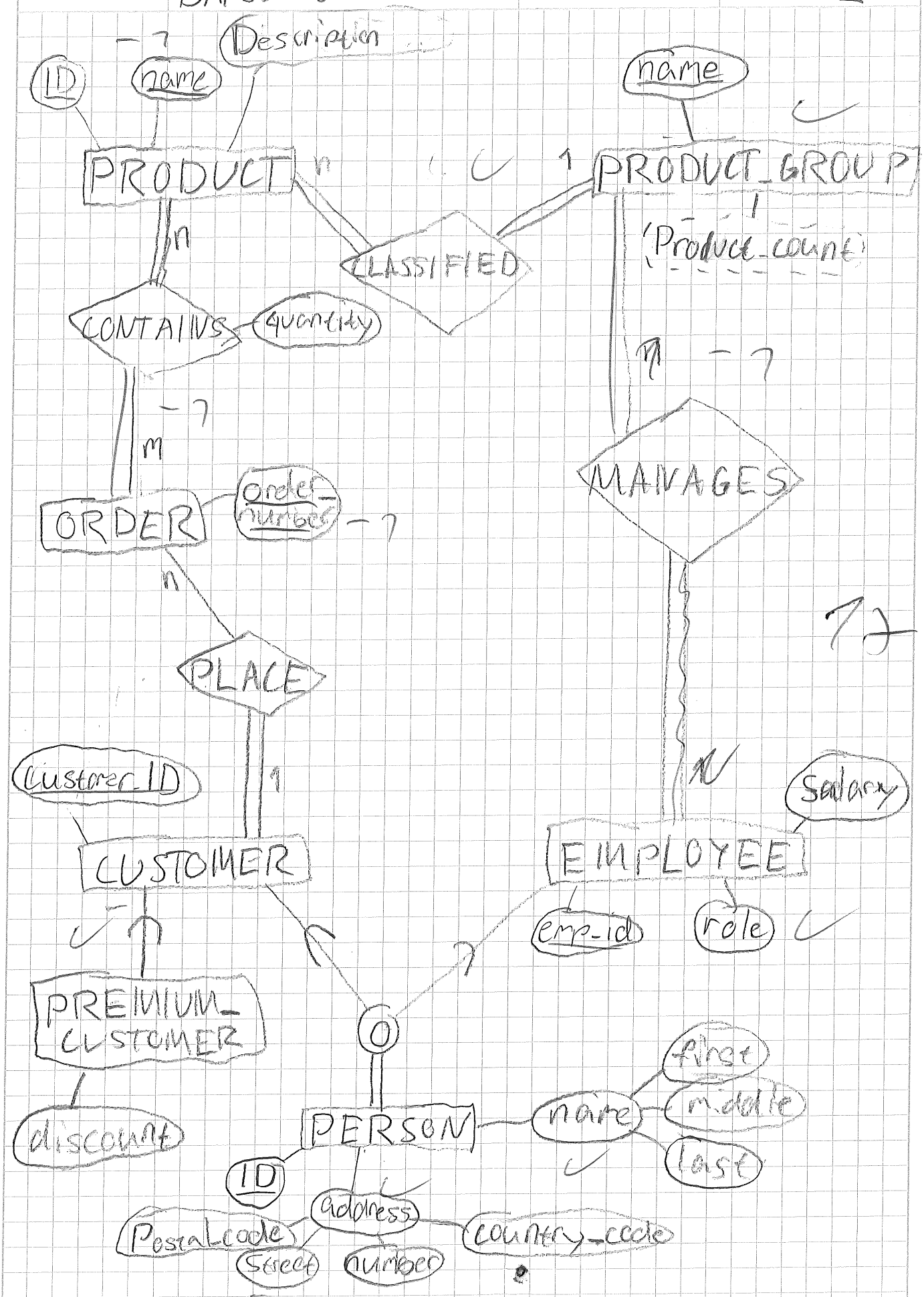
3 A prepared statement is of great help to minimize SQL injection possibilities in the system. + examples

4) Sharding means that different nodes store different data, the data is split up using a "sharding function". This function determines based on the data to be saved which node it shold be persisted to.

A good sharding function splits the data equally between the shards.

5) A JSON document always begins and ends with braces. A minimal JSON document would be {}.
Braces means map and all the content within needs a unique key, "ID" for example. The key is suffixed by a colon (:) and afterwards the content is presented. Content may be strings, numbers, arrays or maps. Strings are noted with " around them and arrays with [].

```
{
  "a": "Hello world!",
  "b": {"id": 0},
  "c": ["a", 1, "c", 0.5]
}
```


ID
name
Description
PRODUCT
n
1
CLASSIFIED
PRODUCT_GROUP
name
Product_count
n
CONTAINS
quantity
m
ORDER
order-number
n
MANAGES
PLACE
Customer_ID
1
CUSTOMER
EMPLOYEE
Salary
emp-id
role
PREMIUM-CUSTOMER
discount
O
PERSON
name
first
middle
last
ID
address
PostalCode
Street
number
country-code

LIBRARY(name, city, country)

SHELF(library, shelf_nr, topic)
library → LIBRARY.name

BOOK(ISBN, title, library, shelf_nr, date_added)
{library, shelf_nr} → {SHELF.library, SHELF.shelf_nr}

FICTION(Book, genre)
Book → BOOK.ISBN

BIOGRAPHY(Book, person)
Book → BOOK.ISBN

BOOK_AUTHORS(Book, author)
Book → BOOK.ISBN

77

2

1)

$\pi_{roomname, capacity}(\sigma_{capacity>50} ROOM)$ ✓ 5

2) $ACME\_BUILDINGS \leftarrow BUILDING \bowtie_{owner=shortname} (\sigma_{full\_name='ACME\ Inc'} COMPANY)$ ✓

$ROOM \bowtie_{building\_name=name,\ building\_city=city} ACME\_BUILDING$ 4

this is not supposed to be indented.

3)

$NO\_BUILDINGS \leftarrow \rho_{(comp, count)}({}_{owner}\mathcal{F}_{count(name, city)}(BUILDING))$ ✓

$\pi_{full\_name, count}(NO\_BUILDINGS \bowtie_{comp=shortname} COMPANY)$ ✓

5

4 $A \leftarrow BUILDING$

$B \leftarrow BUILDING$

$\Pi_{A.name, B.name}(A \bowtie_{A.city = B.city. \; A.owner \neq B.owner.} B)$

↳ name, city is PK!

↳ <>

4

A condition featuring A.name = B.name would have removed duplicates.

1)
```
CREATE TABLE BUILDING(
    name VARCHAR,
    city VARCHAR,
    owner VARCHAR NOT NULL,
    PRIMARY KEY (name, city),
    FOREIGN KEY (owner)
        REFERENCES (COMPANY.shortname)
        ON DELETE CASCADE
);
```
5

2)
```
SELECT * FROM BUILDING
WHERE city IN ('Gothenburg', 'Stockholm',
    'Copenhagen', 'Oslo', 'Helsinki')
ORDER BY owner ASC;
```
5

3)
```
SELECT C.full_name, B.*, R.*
FROM ROOM AS R
INNER JOIN BUILDING AS B
    ON B.name = R.building_name AND
       B.city = R.building_city
RIGHT JOIN COMPANY AS C
    ON B.owner = C.shortname;
```
5

5.5)
```
SELECT roomname
FROM ROOMS
WHERE CAPACITY > ALL(
    SELECT capacity
    FROM ROOM
    WHERE roomname = 'Alfons');
```
✓ ✓ 5

5.4)
```
SELECT AVG(capacity)
FROM ROOM
WHERE building_city = 'Gothenburg';
```
✓

May also use a group by and having clause, shown below, however this is unnecessary.

```
SELECT AVG(capacity)
FROM ROOM
GROUP BY building_city
HAVING building_city = 'Gothenburg';
```
✓ 5