DATABASES

ADIT

Lab compendium – Lab 1 and 2

Institutionen för datavetenskap (IDA), Linköpings universitet

Table of Contents

Lab-Reports

For each lab you will have to hand in a lab-report that includes the name and LiU-id for each group-member as well as SQL-input and the database-response for each question. Also include a short description of what your query does. An example is shown below.

Refer to the course web page for further information and documentation needed for the labs.

Example of Lab-report

```
Lab 1, Anders Andersson(andan123) and Björn Björnsson (bjobj456)

Question 1, Show all tables:
mysql> show tables;
+--------------------+
| Tables_in_dagso62 |
+--------------------+
| jbcity             |
| jbdebit            |
| jbdept             |
| jbemployee         |
| jbitem             |
| jbparts            |
| jbsale             |
| jbstore            |
| jbsupplier         |
| jbsupply           |
+--------------------+
10 rows in set (0.00 sec)

Question 2, Show the fields in jbdept and what type each field is:
mysql> describe jbdept;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| id      | int(11)     | NO   | PRI | 0       |       |
| name    | varchar(20) | YES  |     | NULL    |       |
| store   | int(11)     | NO   | MUL | NULL    |       |
| floor   | int(11)     | YES  |     | NULL    |       |
| manager | int(11)     | YES  | MUL | NULL    |       |
+---------+-------------+------+-----+---------+-------+
5 rows in set (0.01 sec)

Question 3...|
```

The Jonson Brothers' database

The lab exercises in lab 1 and 2 are based on a database that is used for the business of the Jonson Brothers. This section describes that database.
The Jonson Brothers is a retail company with department stores in many major US cities. The company has a large number of employees and sells a varied line of products.
The company consists of a number of stores that contain a number of departments. The company has employees, who (among other things) sell items at the different stores. Sales are registered in the sale and debit tables. Items are bought from various suppliers, who also supply parts for the company's computer equipment. Deliveries of computer parts are registered in the supply table. 计算机部件

Below we provide

- the contents of each table in the database,
- the ER diagram of the database.

Note that all table names are prefixed with JB so as to not enter into conflict with tables created for other courses in your MySQL account.

jbemployee

An employee is identified by an id and described by name, salary, birthyear and startyear. The id of the manager of each employee is also supplied. A null value means that the employee has no manager.

id	name	salary	manager	birthyear	startyear
157	Jones, Tim	12000	199	1940	1960
1110	Smith, Paul	6000	33	1952	1973
35	Evans, Michael	5000	32	1952	1974
129	Thomas, Tom	10000	199	1941	1962
13	Edwards, Peter	9000	199	1928	1958
215	Collins, Joanne	7000	10	1950	1971
55	James, Mary	12000	199	1920	1969
26	Thompson, Bob	13000	199	1930	1970
98	Williams, Judy	9000	199	1935	1969
32	Smythe, Carol	9050	199	1929	1967
33	Hayes, Evelyn	10100	199	1931	1963
199	Bullock, J.D.	27000	NULL	1920	1920
4901	Bailey, Chas M.	8377	32	1956	1975
843	Schmidt, Herman	11204	26	1936	1956
2398	Wallace, Maggie J.	7880	26	1940	1959
1639	Choy, Wanda	11160	55	1947	1970
5119	Bono, Sonny	13621	55	1939	1963
37	Raveen, Lemont	11985	26	1950	1974
5219	Schwarz, Jason B.	13374	33	1944	1959
1523	Zugnoni, Arthur A.	19868	129	1928	1949
430	Brunet, Paul C.	17674	129	1938	1959
994	Iwano, Masahiro	15641	129	1944	1970
1330	Onstad, Richard	8779	13	1952	1971
10	Ross, Stanley	15908	199	1927	1945
11	Ross, Stuart	12067	NULL	1931	1932

jbdept

A department is identified by an id and described by its name as well as its which store and floor it belongs to. The employee id of the manager of the department is also supplied.

id	name	store	floor	manager
35	Book	5	1	55
10	Candy	5	1	13
19	Furniture	7	4	26
20	MajorAppliances	7	4	26
14	Jewelry	8	1	33
43	Children's	8	2	32
65	Junior's	7	3	37
58	Men's	7	2	129
60	Sportswear	5	1	10
99	Giftwrap	5	1	98
1	Bargain	5	0	37
26	Linens	7	3	157
63	Women's	7	3	32
49	Toys	8	2	35
70	Women's	5	1	10
73	Children's	5	1	10
34	Stationary	5	1	33
47	JuniorMiss	7	2	129
28	Women's	8	2	32

jbstore

A store is identified by an id and described by the city it is located within.

id	city
5	941
7	946
8	945
9	941

jbcity

A city is identified by its an id and described by its name and in which state it is located.

id	name	state
900	Los Angeles	Calif
946	Oakland	Calif
945	El Cerrito	Calif
303	Atlanta	Ga
941	San Francisco	Calif
021	Boston	Mass
752	Dallas	Tex
802	Denver	Colo
106	White Plains	Neb
010	Amherst	Mass
981	Seattle	Wash
609	Paxton	Ill
100	New York	NY
921	San Diego	Calif
118	Hickville	Okla
841	Salt Lake City	Utah
537	Madison	Wisc

jbitem

An item is identified by an id and described by its name, the department where it is sold, its price, the quantity on hand (qoh) and the identifier of the supplier that supplied it.

库存

id	name	dept	price	qoh	supplier
26	Earrings	14	1000	20	199
118	Towels, Bath	26	250	1000	213
43	Maze	49	325	200	89
106	Clock Book	49	198	150	125
23	1 lb Box	10	215	100	42
52	Jacket	60	3295	300	15
165	Jean	65	825	500	33
258	Shirt	58	650	1200	33
120	Twin Sheet	26	800	750	213
301	Boy's Jean Suit	43	1250	500	33
121	Queen Sheet	26	1375	600	213
101	Slacks	63	1600	325	15
115	Gold Ring	14	4995	10	199
25	2 lb Box, Mix	10	450	75	42
119	Squeeze Ball	49	250	400	89
11	Wash Cloth	1	75	575	213
19	Bellbottoms	43	450	600	33
21	ABC Blocks	1	198	405	125
107	The `Feel' Book	35	225	225	89
127	Ski Jumpsuit	65	4350	125	15

jbdebit

A debit (receipt of a sale) is identified by its id and described by the timestamp sdate when the debit took place, the employee who sold the item, and a customer account to which the amount was debited.

id	sdate	employee	account
100581	15-JAN-95 12:06:03	157	10000000
100582	15-JAN-95 17:34:27	1110	14356540
100586	16-JAN-95 13:53:55	35	14096831
100592	17-JAN-95 09:35:23	129	10000000
100593	18-JAN-95 12:34:56	35	11652133
100594	19-JAN-95 10:10:10	215	12591815

jbsale

Each debit can contain a number of items, each represented as a sale. Each sale is identified by the debit to which it belongs and the id of the item that was sold and also describes the quantity of items sold. For example: Debit transaction 100581 consists of two items: item 118 with a quantity of 5 and item 120 with quantity 1.

debit	item	quantity
100581	118	5
100581	120	1
100582	26	1
100586	127	3
100586	106	2
100592	258	1
100593	23	2
100594	52	1

jbsupplier

A supplier (of items and parts) is identified by its id and described by its name and the city in which it is located.

id	name	city
199	Koret	900
213	Cannon	303
33	Levi-Strauss	941
89	Fisher-Price	021
125	Playskool	752
42	Whitman's	802
15	White Stag	106
475	DEC	010
122	White Paper	981
440	Spooley	609
241	IBM	100
62	Data General	303
5	Amdahl	921
20	Wormley	118
67	Edger	841
999	A E Neumann	537

jbparts

A part, used internally by the store, not sold to customers, is identified by its id and described by its name, color, weight, and the quantity on hand (qoh).

id	name	color	weight	qoh
1	central processor	pink	10	1
2	memory	gray	20	32
3	disk drive	black	685	2
4	tape drive	black	450	4
5	tapes	gray	1	250
6	line printer	yellow	578	3
7	l-p paper	white	15	95
8	terminals	blue	19	15
13	paper tape reader	black	107	0
14	paper tape punch	black	147	0
9	terminal paper	white	2	350
10	byte-soap	clear	0	143
11	card reader	gray	327	0
12	card punch	gray	427	0

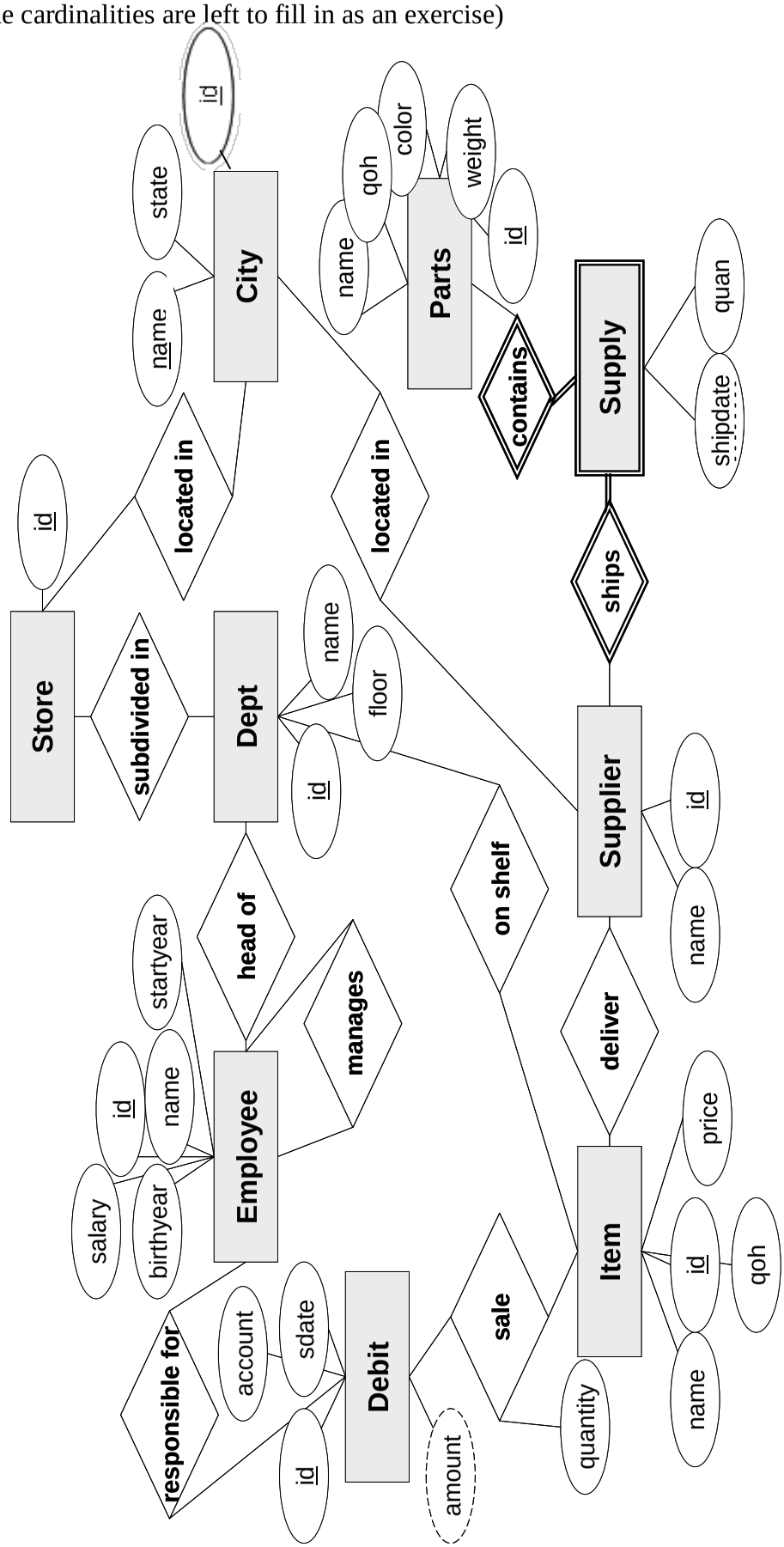
jbsupply

A supplier supplies the different parts. Each part is supplied separately (even though they may be transported together) and each supply is identified by the supplier, the part id and the date it was shipped. It quantity of items supplied in each supply is also described.

supplier	part	shipdate	quan
475	1	1993-12-31	1
475	2	1994-05-31	32
475	3	1993-12-31	2
475	4	1994-05-31	1
122	7	1995-02-01	144
122	7	1995-02-02	48
122	9	1995-02-01	144
440	6	1994-10-10	2
241	4	1993-12-31	1
62	3	1994-06-18	3
475	2	1993-12-31	32
475	1	1994-07-01	1
5	4	1994-11-15	3
5	4	1995-01-22	6
20	5	1995-01-10	20
20	5	1995-01-11	75
241	1	1995-06-01	1
241	2	1995-06-01	32
241	3	1995-06-01	1
67	4	1995-07-01	1
999	10	1996-01-01	144
241	8	1995-07-01	1
241	9	1995-07-01	144
89	3	1995-07-04	1000
89	4	1995-07-04	1000

The Jonson Brothers' ER-Diagram

(The cardinalities are left to fill in as an exercise)



Lab 1; SQL-Queries and Views

Objectives

The purpose of this exercise is to practise writing queries in SQL, including the use of aggregate functions and views.

Background Reading

Check the lecture material and study the SQL chapter(s) in your database book. Note that small discrepancies might exist between some SQL interpreters and some books, as they follow slightly different SQL standards.

Setting up your environment and database

To set up your computer and load the database on which you will work for the two coming labs, follow the instructions available at the course web page.

The Lab

Formulate in SQL.

1) List all employees, i.e. all tuples in the ***jbemployee*** relation.
2) List the name of all departments in alphabetical order. Note: by "name" we mean the name attribute for all tuples in the ***jbdept*** relation.
3) What parts are not in store, i.e. qoh = 0? (qoh = Quantity On Hand)
4) Which employees have a salary between 9000 (included) and 10000 (included)?
5) What was the age of each employee when they started working (startyear)?
6) Which employees have a last name ending with "son"?
7) Which items (note **items**, not parts) have been delivered by a supplier called *Fisher-Price*? Formulate this query using a subquery in the where-clause.
8) Formulate the same query as above, but without a subquery.
9) Show all cities that have suppliers located in them. Formulate this query using a subquery in the where-clause.
10) What is the name and color of the parts that are heavier than a card reader? Formulate this query using a subquery in the where-clause. (The SQL query must not contain the weight as a constant.)
11) Formulate the same query as above, but without a subquery. (The query must not contain the weight as a constant.)
12) What is the average weight of black parts?
13) What is the total weight of all parts that each supplier in Massachusetts ("Mass") has delivered? Retrieve the name and the total weight for each of these suppliers. Do not forget to take the quantity of delivered parts into account. Note that one row should be returned for each supplier.

14) Create a new relation (a table), with the same attributes as the table items using the CREATE TABLE syntax where you define every attribute explicitly (i.e. not as a copy of another table). Then fill the table with all items that cost less than the average price for items. Remember to define primary and foreign keys in your table!

15) Create a view that contains the items that cost less than the average price for items.

16) *What is the difference between a table and a view?* One is static and the other is dynamic. Which is which and what do we mean by static respectively dynamic?

17) Create a view, using only the implicit join notation, i.e. only use *where* statements but no *inner join*, *right join* or *left join* statements, that calculates the total cost of each debit, by considering price and quantity of each bought item. (To be used for charging customer accounts). The view should contain the sale identifier (debit) and total cost.

18) Do the same as in (17), using only the explicit join notation, i.e. using only *left*, *right* or *inner* joins but no *where* statement. Motivate why you use the join you do (left, right or inner), and why this is the correct one (unlike the others).

19) Oh no! An earthquake!

    a) Remove all suppliers in Los Angeles from the table ***jbsupplier***. This will not work right away (you will receive error code 23000) which you will have to solve by deleting some other related tuples. However, do not delete more tuples from other tables than necessary and do not change the structure of the tables, i.e. do not remove foreign keys. Also, remember that you are only allowed to use “Los Angeles” as a constant in your queries, not “199” or “900”.

    b) Explain what you did and why.

20) An employee has tried to find out which suppliers that have delivered items that have been sold. He has created a view and a query that shows the number of items sold from a supplier.

```
mysql> CREATE VIEW jbsale_supply(supplier, item, quantity) AS
    -> SELECT jbsupplier.name, jbitem.name, jbsale.quantity
    -> FROM jbsupplier, jbitem, jbsale
    -> WHERE jbsupplier.id = jbitem.supplier
    -> AND jbsale.item = jbitem.id;
Query OK, 0 rows affected (0.01 sec)

mysql> SELECT supplier, sum(quantity) AS sum FROM jbsale_supply
    -> GROUP BY supplier;
+--------------+---------------+
| supplier     | sum(quantity) |
+--------------+---------------+
| Cannon       |             6 |
| Levi-Strauss |             1 |
| Playskool    |             2 |
| White Stag   |             4 |
| Whitman's    |             2 |
+--------------+---------------+
5 rows in set (0.00 sec)
```

The employee would also like include the suppliers which has delivered some items, although for whom no items have been sold so far. In other words he wants to list all suppliers, which has supplied any item, as well as the number of these

items that have been sold. Help him! Drop and redefine `jbsale_supply` to consider suppliers that have delivered items that have never been sold as well. **Hint:** The above definition of `jbsale_supply` uses an (implicit) inner join that removes suppliers that have not had any of their delivered items sold.

Handing in

- For each problem in the lab, list the query followed by the result of the query.
- Written answers for questions *(16), 18* and (*19b)*.

Lab 2; Database Design and Entity-Relationship Modelling

Objectives

The purpose of this exercise is to give a good understanding of database design and entity-relationship modelling.

Background Reading

Read lecture material on ER- and EER-modelling, on the translation of EER-diagrams into relational tables, and SQL for creating tables and managing constraints.

The Lab

The Jonson Brothers' business is expanding and the database is continuously being extended with new information. The current state of the company database can be seen in the ER-diagram provided on page 8. The management of Jonson Brothers' has hired you to help them extend their database. The work requires extensions to support a bonus system where managers can be given an extra bonus (e.g. if their department have met their sale predictions). The management also wants to encourage customers to shop more by creating a credit card that users can use when paying for items that they buy.

1) **(At home)** Analyze the ER-diagram on page 8 (available electronically from the course web page) and the relational database and add information about the cardinality of the relationships such as one-to-one, one-to-many, many-to-one, and many-to-many to the ER-diagram.

2) **(At home)** Extend the ER-diagram with an entity type *manager* that is a subclass of *employee*. Add support for a manager bonus that is added to the salary by giving the manager entity a bonus attribute. Use the manager-entity (instead of employee) in appropriate, existing relationships. Note that some managers are managers of departments, some managers are managers of other employees, and some are both. Draw your extensions to the ER-diagram and translate them to the relational model.

3) Implement your extensions in the database by first creating tables, if any, then populating them with existing manager data, then adding/modifying foreign key constraints. *Do you have to initialize the bonus attribute to a value? Why?*

4) All departments showed good sales figures last year! Give all current department managers $10,000 in bonus. This bonus is an addition to other possible bonuses they have.
   Hint: Not all managers are department managers. Update all managers that are referred in the ***jbdept*** relation.

5) **(Partly at home)** In the existing database, customers can buy items and pay for them, as reflected by the sale and debit tables. Now, you want to create support for storing customer information. The customers will have accounts, where they can deposit and withdraw money. The requirements are:
    - Customers are stored with name, street address, city, and state. Use existing city information!
    - A customer can have several accounts.
    - Accounts are stored with account number, and balance.
    - Information about transactions such as the type (withdrawal/deposit/sale), transaction number, account number, date and time of the transaction, amount, and the employee responsible for the transaction (that is, the employee that registers the transaction, not the customer that owns the account) should be stored in a transaction-entity. Use subclasses to distinguish between the different types of transactions (withdrawals/deposits/sales). This means that the new transaction entity will be a superclass of the existing debit-entity.

    a) **(At home)** Extend the EER-diagram with your new entities, relationships, and attributes.

    b) Implement your extensions in your database. Add primary key constraints, foreign key constraints and integrity constraints to your table definitions. Do not forget to correctly set up the new and existing foreign keys.

- Use `alter table t1 drop foreign key constraint_name;` and
  ```
  alter table t2
  add constraint constraint_name
  foreign key (t2_attribute) references t1(t1_attribute);
  ```
  to change existing foreign keys.
- You may delete table data from the ***jbsale***-table in order to properly link the new transaction-table into the existing schema.

Handing in

- ER diagram with *cardinalities* and the *extensions* required by questions *2)* and *5.a).*
- Relational model corresponding to both ER diagram and the implemented database.
- The SQL commands to create, modify and delete tables and constraints (such as foreign keys), as well as the results of these commands.
- A written answer for question *3).*

You are now ready to work with the EER-diagram part and the EER-diagram-to-table translation part of the project.