

DEPARTMENT OF COMPUTER AND INFORMATION SCIENCE (IDA)
DIVISION FOR DATABASE AND INFORMATION TECHNIQUES (ADIT)

Database Technologies

Assignment: SQL and Views

Objective

The objective of this assignment is to practice using the database language SQL, including the use of aggregation functions and views.

Background Reading

Lecture material and book chapters about SQL. Note that small discrepancies might exist between some SQL interpreters and some books, as they follow slightly different SQL standards.

Introduction

This assignment is based on an existing database of a retail company called The Jonson Brothers. For an overview of this database, see the appendix of this document. The tables of this database have to be imported into your database on the database server provided by the university. For instructions on how to access that database server and how to import the Jonson Brothers database, refer to the course Website.

Tasks

Write an SQL statement for each of the following points.

1. List all employees, i.e., all tuples in the *jbemployee* relation. *¹
2. List the name of all departments in alphabetical order. Note: by “name” we mean the name attribute in the *jbdept* relation.
3. What parts are not in store? Note that such parts have the value 0 (zero) for the *qoh* attribute (*qoh* = quantity on hand). *
4. List all employees who have a salary between 9000 (included) and 10000 (included)? *
5. List all employees together with the age they had when they started working? Hint: use the *startyear* attribute and calculate the age in the SELECT clause. *
6. List all employees who 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 by using a subquery in the WHERE clause.
8. Formulate the same query as above, but without a subquery. *

¹ All questions marked with an asterisk (*) can be tackled already after the first SQL-related lecture session. That is, the features of SQL that you have learned about at this point are enough for these questions.

9. List all cities that have suppliers located in them. Formulate this query using a subquery in the WHERE clause.
10. What is the **name** and the **color** of the parts that are **heavier** than a **card reader**? Formulate this query using a subquery in the WHERE clause. (The query must not contain the weight of the card reader as a constant; instead, the weight has to be retrieved within the query.)
11. Formulate the same query as above, but without a subquery. Again, the query must not contain the weight of the card reader as a constant.
12. What is the average weight of all black parts?
13. For every supplier in Massachusetts ("Mass"), retrieve the name and the total weight of all parts that the supplier has delivered? 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 with the same attributes as the **jbitems** relation by using the CREATE TABLE command where you define every attribute explicitly (i.e., not as a copy of another table). Then, populate this new relation with all items that **cost less than the average price** for all items. Remember to define the **primary key 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 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 the **total cost**. In the query that defines the view, capture the join condition in the **WHERE clause** (i.e., do *not* capture the join in the FROM clause by using keywords *inner join*, *right join* or *left join*).
18. Do the same as in the previous point, but now capture the join conditions in the FROM clause by using only *left*, *right* or *inner* joins. Hence, the WHERE clause must not contain any join condition in this case. Motivate why you use type of join you do (left, right or inner), and why this is the correct one (in contrast to the other types of joins).
19. Oh no! An earthquake!
 - a) Remove all suppliers in Los Angeles from the *jbsupplier* table. This will not work right away. Instead, you will receive an error with 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, 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** have delivered items that have been sold. To this end, the employee has created a view and a query that lists 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)
```

Now, the employee also wants to include the suppliers that have delivered some items, although for whom no items have been sold so far. In other words, he wants to list all suppliers that have supplied any item, **as well as the number of these items that have been sold**. Help him! Drop and redefine the *jbsale_supply* view to also consider suppliers that have delivered items that have never been sold.

Hint: Notice that 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

Hand in an executable SQL file that contains each of the SQL statements and, in comments, the output produced by executing the statement as well as the text answers for points 16, 18, and 19b. Additionally, the file should start with a comment that contains your names (yours and your lab partner's) as well as your LiU IDs. Hence, this file should look something like the following.

```
/*Lab 1, Anders Andersson (andan123) and Björn Björnsson (bjobj456)*/
SOURCE company_schema.sql;
SOURCE company_data.sql;
```

```
/*Question 1: Print a message that says "hello world"*/
SELECT 'hello world!' AS 'message';
```

```
/*
+-----+
| message |
+-----+
| hello world! |
+-----+
1 row in set (0.00 sec)
*/
```

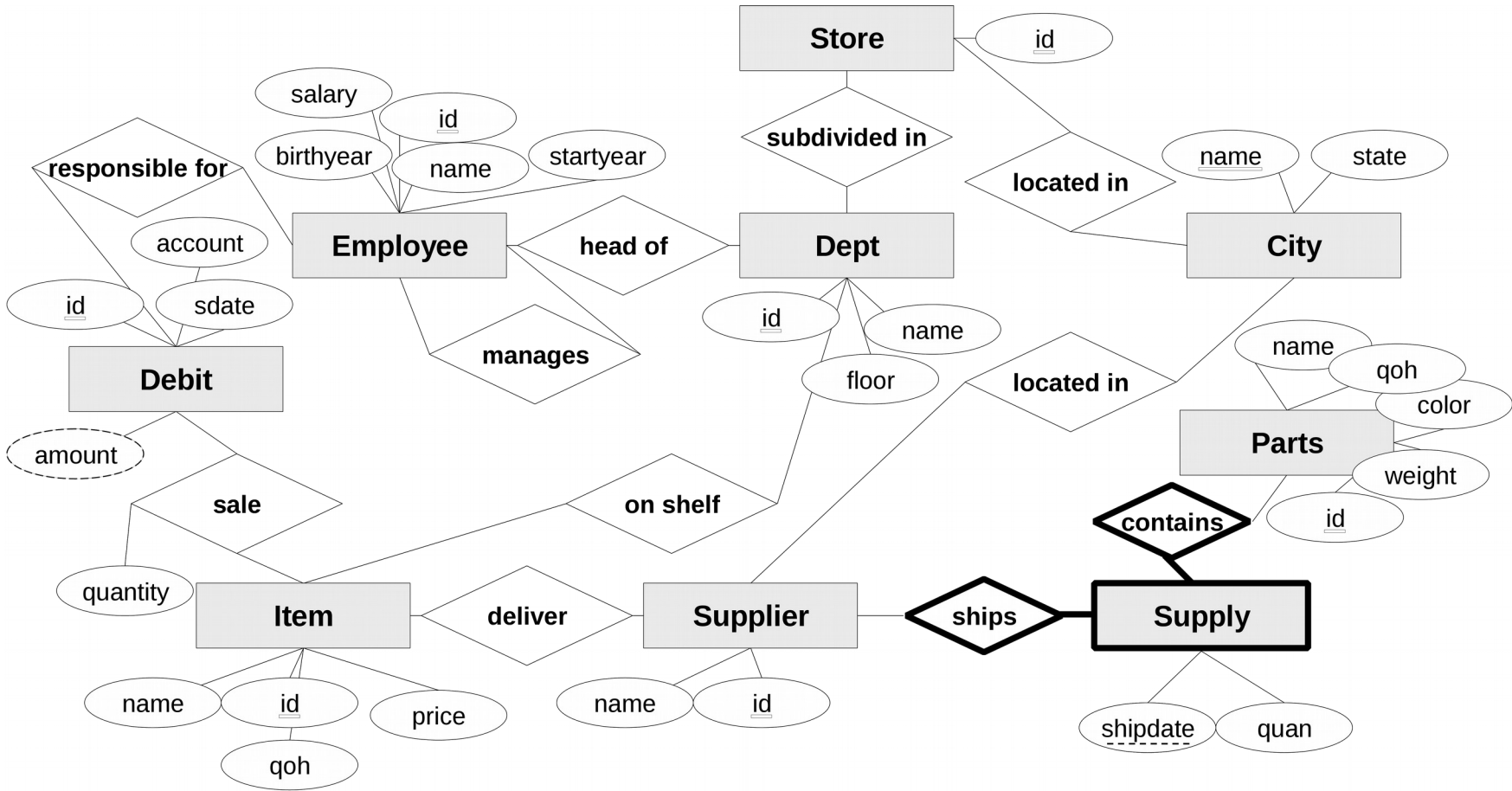
```
/* Question 2: List all tables */
SHOW TABLES;
```

```
/*
+-----+
| Tables_in_andan123 |
+-----+
| jbcity              |
| jbdebit             |
| jbdept              |
| jbemployee          |
| jbitem              |
| jbparts             |
| jbsale              |
| jbstore             |
| jbsupplier          |
| jbsupply            |
+-----+
10 rows in set (0.00 sec)
*/
```

```
/* Question 3: What does the acronym SQL stand for? */
/* Answer: SQL stands for Structured Query Language. */
```

Appendix: The Jonson Brothers Database

The Jonson Brothers is a retail company with department stores in many major US cities. Each store contains several departments. The company has employees who (among other things) sell items at the different stores. Sales are registered in the sale and the debit tables of the company's database. 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 of the database. This appendix provides an overview of this database, beginning with an ER diagram of the types of entities and their relationships as captured in the database (see next page). Thereafter, the various tables in the relational schema of the database are described.



The remainder of this appendix describes the tables of the relational database schema of the Jonson Brothers database. All table names are prefixed with *jb* to avoid conflicts with tables created for other courses in your MySQL account.

jbcity: A city is identified by an id and has a name and the state it is located in.

<u>id</u>	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

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

<u>id</u>	city
5	941
7	946
8	945
9	941

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

<u>id</u>	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

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.

<u>id</u>	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.

<u>id</u>	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. Its quantity of items supplied in each supply is also described.

<u>supplier</u>	<u>part</u>	<u>shipdate</u>	<u>quan</u>
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

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.

<u>id</u>	<u>name</u>	<u>salary</u>	<u>manager</u>	<u>birthyear</u>	<u>startyear</u>
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