The Jonson Brothers' database

Some lab exercises are based on a database which is used for the business of the Jonson Brothers. This section describes this 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. To manage all information about the company structure and products, a database system is used. The company consists of a number of *stores* that each contain a number of *departments*. The company has a number of *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.

The *sale* and *debit* tables may be a bit tricky to understand. You can view a row in the *debit* table as representing the receipt you get when you pay for your items, while a row in the *sale* table represents a row on such a receipt.

The following gives

- The content of each table in the database (i.e. the state of the data before you do any modification).
- The ER diagram of the database

1. The content of the tables.

```
SELECT * FROM employee;
25 rows selected
```

KEY	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		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		1931	1932

```
SELECT * FROM dept;
19 rows selected
```

KEY	NAME	STORE	FLOOR	MANAGER
35	Book	5	1	55
10	Candy	5	1	13
19	Furniture	7	4	26
20	Major Appliances	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	Junior Miss	7	2	129
28	Women's	8	2	32

```
SELECT * FROM item;
20 rows selected
```

KEY	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

```
SELECT * FROM parts;
14 rows selected
```

KEY	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

```
SELECT * FROM sale;
8 rows selected
```

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

```
SELECT * FROM debit;
6 rows selected
```

KEY	SDATE	EMPLOYEE	ACCOUNT
100581	15-JAN-95	157	10000000
100582	15-JAN-95	1110	14356540
100586	16-JAN-95	35	14096831
100592	17-JAN-95	129	10000000
100593	18-JAN-95	35	11652133
100594	19-JAN-95	215	12591815

```
SELECT * FROM city;
17 rows selected
```

NAME	STATE
Los Angeles	Calif
Oakland	Calif
El Cerrito	Calif
Atlanta	Ga
San Francisco	Calif
Boston	Mass
Dallas	Tex
Denver	Colo
White Plains	Neb
Amherst	Mass
Seattle	Wash
Paxton	Ill
New York	NY
San Diego	Calif
Hickville	Okla
Salt Lake City	Utah
Madison	Wisc

```
SELECT * FROM store;
4 rows selected
```

KEY	CITY
5	San Francisco
7	Oakland
8	El Cerrito
9	San Francis

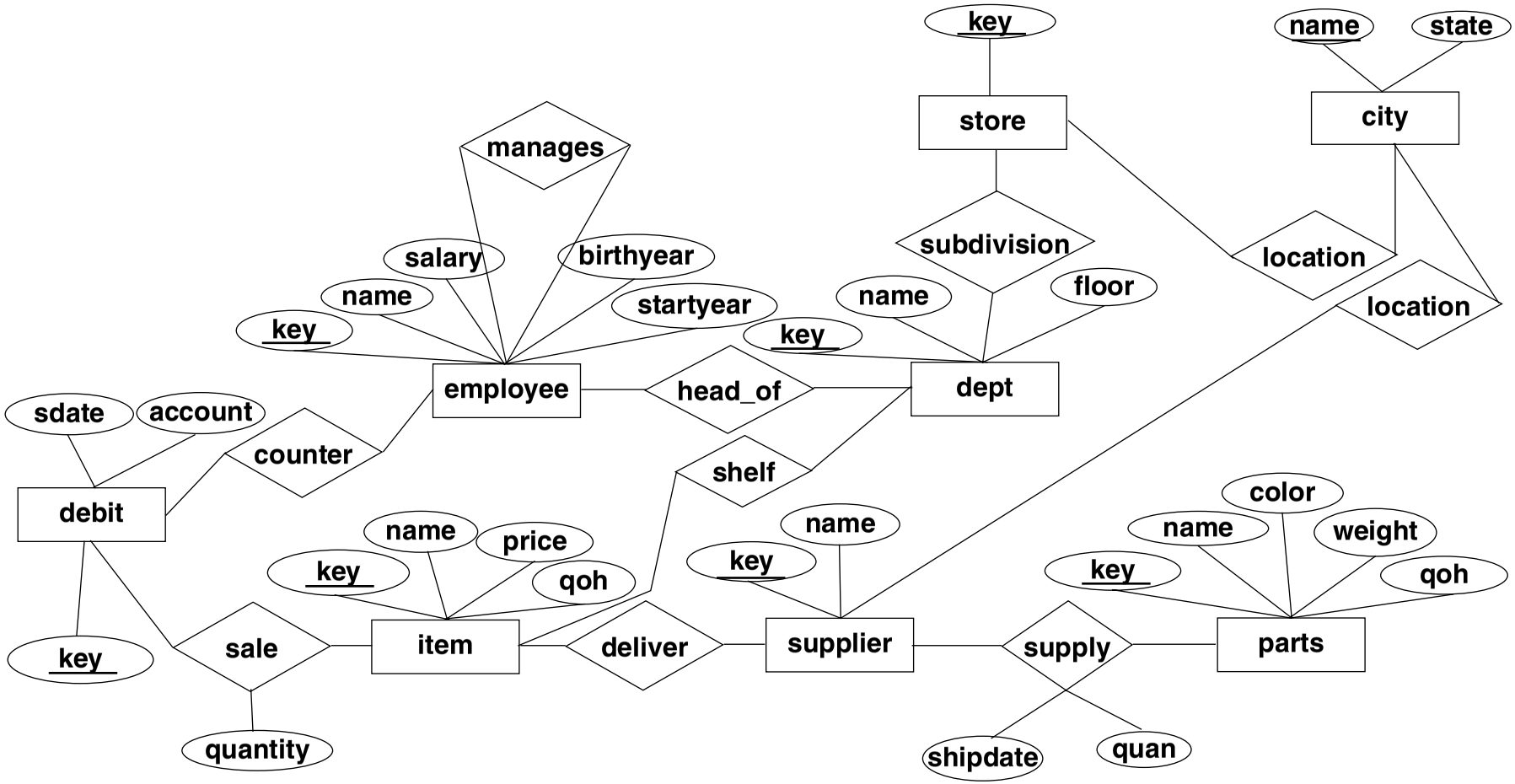
```
SELECT * FROM supply;
25 rows selected
```

SUPPLIER	PART	SHIPDATE	QUAN
475	1	1993-12-31 00:00:00	1
475	2	1994-05-31 00:00:00	32
475	3	1993-12-31 00:00:00	2
475	4	1994-05-31 00:00:00	1
122	7	1995-02-01 00:00:00	144
122	7	1995-02-02 00:00:00	48
122	9	1995-02-01 00:00:00	144
440	6	1994-10-10 00:00:00	2
241	4	1993-12-31 00:00:00	1
62	3	1994-06-18 00:00:00	3
475	2	1993-12-31 00:00:00	32
475	1	1994-07-01 00:00:00	1
5	4	1994-11-15 00:00:00	3
5	4	1995-01-22 00:00:00	6
20	5	1995-01-10 00:00:00	20
20	5	1995-01-11 00:00:00	75
241	1	1995-06-01 00:00:00	1
241	2	1995-06-01 00:00:00	32
241	3	1995-06-01 00:00:00	1
67	4	1995-07-01 00:00:00	1
999	10	1996-01-01 00:00:00	144
241	8	1995-07-01 00:00:00	1
241	9	1995-07-01 00:00:00	144
89	3	1995-07-04 00:00:00	1000
89	4	1995-07-04 00:00:00	1000

```
SELECT * FROM supplier;
16 rows selected
```

KEY	NAME	CITY
199	Koret	Los Angeles
213	Cannon	Atlanta
33	Levi-Strauss	San Francisco
89	Fisher-Price	Boston
125	Playskool	Dallas
42	Whitman's	Denver
15	White Stag	White Plains
475	DEC	Amherst
122	White Paper	Seattle
440	Spooley	Paxton
241	IBM	New York
62	Data General	Atlanta
5	Amdahl	San Diego
20	Wormley	Hickville
67	Edger	Salt Lake City
999	A E Neumann	Madison

2. the E/R diagram



Don’t forget to add the cardinality quotas (“1:N” etc.)!

Exercise 1: SQL - Queries and Views (Sun lab)

1.1 Objectives

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

1.2 Time Needed

You should complete this lab within 90-120 minutes.

1.3 Background reading

Read these instructions, check the lecture material and study the SQL chapter(s) in your database book. Note that not all SQL interpreters always follow the standard. The course web page gives more hints for this.

1.4 Setting up your computer and your database

To set up your computer and load the database on which you will work for the three coming labs, follow the instructions from the page “Labs - Settings” on the web page of the course.

1.5 The lab

Use SQL to find the answers to the questions below. Your lab report should show how you formulated the queries AND the answers you got.

1) List all employees, i.e. all tuples in the EMPLOYEE relation!
2) List the name of all departments in alphabetical order. Note: by “name” we mean the NAME attribute for all tuples in the DEPT relation.
3) What parts are not in store, i.e. QOH=0? (QOH = Quantity On Hand)
4) Which employees have a salary between 9000 and 10000?
5) What was the age of each employee when they started working (STARTYEAR) here?
6) Which employees have a last name ending with “son”?
7) Which items 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) 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.)
10) Formulate the same query as above, but without a subquery. (The query must not contain the weight as a constant.)
11) What is the average weight of black parts?
12) 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. Don’t forget

to take the quantity of delivered parts into account.

13) Create a new relation (a table) that contains the items that cost less than the average price for items! Do not forget to define keys in your table!

14) Create a view that contains the items that cost less than the average price for items! *What is the difference between (13) and (14)?*

15) Create a view that calculates the total cost of each sale, by considering price and quantity of each bought item. (To be used for charging customer accounts). Should return the sale identifier (debit) and total cost.

16) Oh no! An earthquake! Remove all suppliers in Los Angeles from the table SUPPLIERS. *What happens when you try to do this? Why? Explain the specific error message. Which tuples make the delete-statement fail?*

17) A database manager in the company has tried to find out what suppliers have delivered items that have been sold. He has created a help view and can find how many items sold from each supplier of the items:

```
SQL> create view sale_supply(supplier, item, quantity) as
         select supplier.name, item.name, sale.quantity
         from supplier, item, sale
         where supplier.key = item.supplier and
               sale.item = item.key;
SQL> select supplier,sum(quantity) from sale_supply
group by supplier;
6 rows selected
----------------------------------
|      SUPPLIER      |    SUM    |
----------------------------------
|       Cannon       |     6     |
|       Koret        |     1     |
|    Levi-Strauss    |     1     |
|     Playskool      |     2     |
|     White Stag     |     4     |
|     Whitman's      |     2     |
----------------------------------
SQL>
```

One problem is that he also would like to find suppliers that have not had any of their delivered items sold. Help him! Drop and redefine `sale_supply` to consider suppliers that have delivered items that have never been sold as well.
Hint: The above definition of `sale_supply` uses an (implicit) *inner join* that removes suppliers that have not had any of their delivered items sold.

1.6 Handing in

- A test run of your queries (i.e. give the query followed by the result of the query).
- Clear answers for questions 14 and 16.

Exercise 2: Database Design and E/R Modeling (Sun lab)

2.1 Objectives

The purpose of this exercise is to give a good understanding of database design and Entity/Relationship modeling.

2.2 Time Needed

You should complete this lab in 90-120 minutes.

2.3 Background reading

Read lecture material on E/R and EE/R modeling, on the translation of EE/R into relational tables, and SQL for creating tables and managing constraints.

2.4 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 E/R diagram provided in the "Jonson Brothers' database" section. 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) added to their salary. 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.

2.4.1 Exercises

1) **(At home)** Start by analyzing the E/R diagram and the relational database. Then add information about the cardinality of the relationships such as one-to-one, one-to-many, many-to-one, and many-to-many.
2) **(Partly at home)** Extend the E/R diagram with an entity type *manager* that is a sub-class 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. Draw your extensions to the E/R diagram, translate to the relational model, and implement the extension in the company database.
3) Now that you have changed the schema, also change the data, so that all managers are managers! That is, if you have made a manager table, you should insert data in it. Note that some managers are managers of departments, some managers are managers of other employees, and some are both. *Do you have to initialize the bonus to a value? Why?*
4) All departments showed good sales figures last year! Give all current department managers 10000 in bonus. This bonus is an addition to possible other bonuses they have. **Hint:** Not all managers are department managers. Update all managers that exist in the *dept* relation. Note: commit!
5) **(Partly at home)** In the existing database, your customers can buy things and pay for them, as reflected by the *sale* and *debit* tables. Now, you want to create support for a customer card, with possible credit. The customers will have accounts, where they can de-

posit and withdraw money. The requirements are:

- Information must be stored about *customers* such as name, street address, city, and state. Note: several customers can have the same name and address.
- Information about *accounts* such as account number, balance, and allowed credit. Do not put any constraints on balance and credit.
- Information about *transactions* (withdrawals/deposits/sales) such as transaction number, account number, date and time of deposit/withdrawal/sale, amount, and the employee responsible for the transaction (that is, the employee that registers the transaction, not the customer that owns the account). Thus, *transaction* effectively replaces/extends the existing *debit*-entity.
- A customer can have several accounts.

Extend the E/R diagram with your new entities, relations, and attributes **(at home)**. Implement your extension in your database. Add primary key and foreign key constraints to your table definitions. Do not forget to correctly set up the new and existing foreign keys.

Hints:

- Use `alter table t1 drop constraint 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 sale-table in order to properly link the new transaction-table into the existing schema (check that you have passed the first lab before doing so).
- **Read the SQL hints for this lab on the course web page.**

2.5 Handing in

- ER diagram with cardinalities on the relations, and the extensions required by question 2) and 5). Note that you can get a copy of the original ER diagram at the link “Labs - E/R diagram” in the web page of the course.
- The test run of your queries when you change the relational schema., i.e. 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.