

Database Implementation Lab

Installation

1. RDBMS Installation: Install the DBMS software as per the instructions specified in the product document. For MySQL, you can find the instructions here: <http://dev.mysql.com/doc/refman/5.1/en/installing.html>
2. Client Installation: All RDBMS can work with a variety of client tools. Some client tools provide a graphical user interface to the database. You must NOT use any GUI clients for your activities. For MySQL, everyone must be familiar and be comfortable using the command line tool called "mysql".

Note: You must complete the installation BEFORE coming to the lab session.

Exercise #1

The aim of the lab exercise is to implement the Company Database schema ("company_handout.pdf") using the database definition guidelines and database manipulation guidelines given in this document. As part of the implementation, you must create and demonstrate the following SQL scripts that follow the given guidelines:

1. create_companydb.sql
2. alter_companydb.sql
3. drop_companydb.sql
4. insert_companydb.sql
5. update_comanydb.sql

Exercise #2

Create SQL scripts for the following queries (one script for each query):

1. Start with a few generic queries of your choice from the schema
2. Write a query to display the names of projects that Joyce English is working on.
3. Write a query to display the name of the employee along with the employee's salary and the supervisor's salary. (Do NOT use nested queries)
4. Write a query to display the names of all the employees who are on bench (i.e., who are not working on any project).
5. Show the result (NOT the query) of "EMPLOYEE left outer join DEPENDENT" (display only the key values from each of the tables).
6. Write a query to display the ESSN of the employee who has clocked the maximum project hours.
7. Write a query to show the name of employees who work on all the projects.

Database Definition Guidelines

1. First create an empty database using the procedure appropriate for the respective DBMS. For example, in MySQL, use "CREATE DATABASE" statement.
2. Have the relational schema design ready on paper. Having it in tabular form that we discussed will make it easier.
3. Full documentation on SQL DDL can be found here: <http://dev.mysql.com/doc/refman/5.1/en/sql-syntax-data-definition.html>
4. The relation schema design must be implemented using a combination of three DDL script files as noted below:
 - a. "CREATE" scripts: Create ONE large script file called **create_dbname.sql** (where dbname is replaced by the short name of the database; e.g., create_companydb.sql). This script file should contain:
 - i. Basic CREATE TABLE statements for each table in the schema
 - ii. PK Constraint against the appropriate attribute
 - iii. NOT NULL constraints as applicable attributes
 - iv. Do NOT include any FK constraints yet.

Example for one table:

```
create table department(
    dname varchar(30),
    dnumber smallint,
    mgr_ssn char(6),
    mgr_start_date date,
    constraint pk_department PRIMARY KEY (dnumber)
);
```

Full documentation on CREATE TABLE syntax can be found here:

- b. "ALTER" scripts: Create ONE large script file called alter_dbname.sql (e.g., alter_companydb.sql). This script file should contain:
 - i. ALTER TABLE statements to add foreign key constraints for each table as applicable.
 - ii. ALTER TABLE statements to add any further semantic constraints as applicable.

Example for one table:

```
alter table employee
    add constraint fk_super_ssn FOREIGN KEY (super_ssn)
REFERENCES employee(ssn);
```

- c. "DROP" scripts: Create ONE large script file called drop_dbname.sql (e.g., drop_companydb.sql). This script file should contain:
 - i. ALTER TABLE statements to drop each of the FKs that were added.

Example for one table:

```
alter table employee
    drop FOREIGN KEY fk_super_ssn;
```

- ii. DROP TABLE statements to drop each of the tables.

Example for one table:
`drop table employee;`

Database Manipulation Guidelines

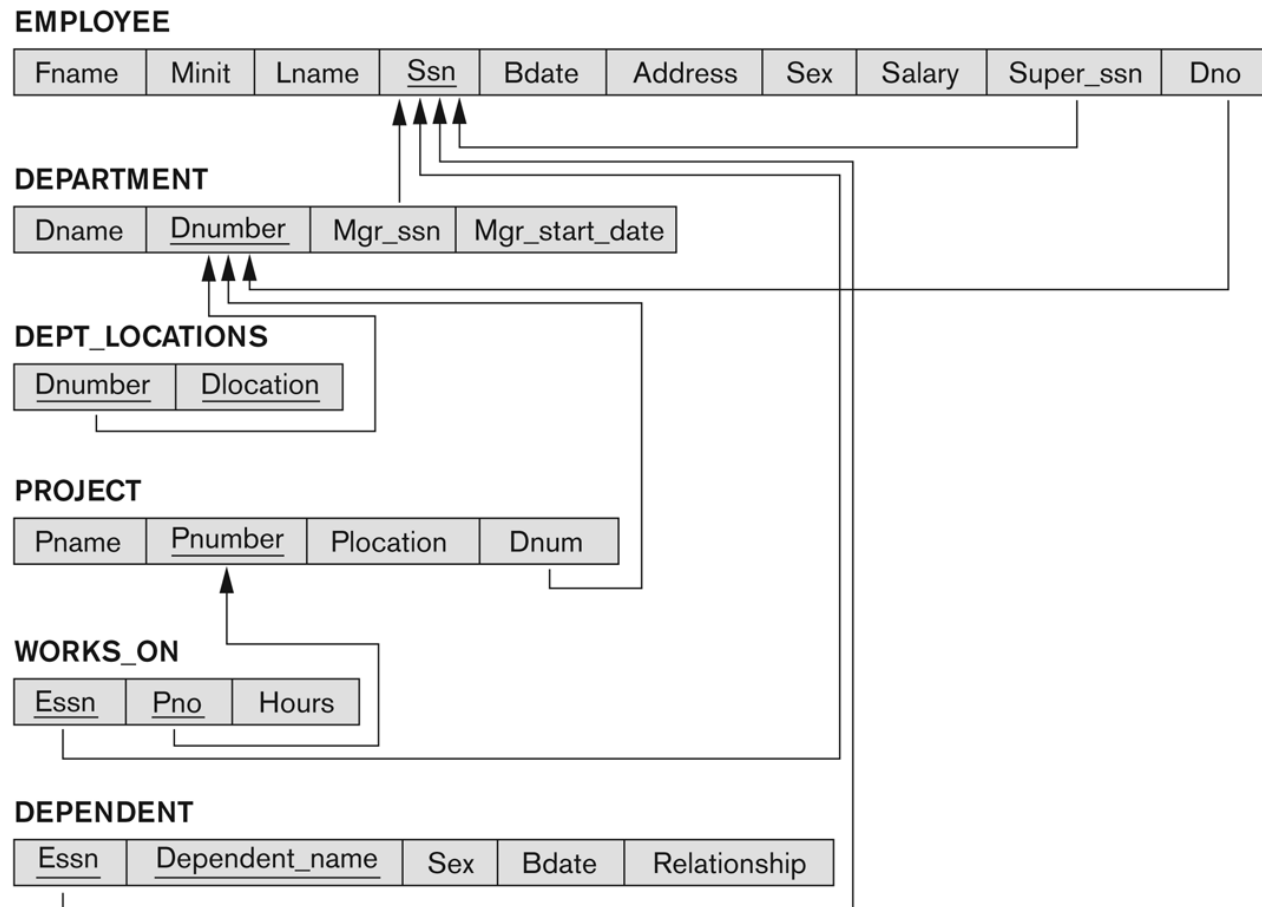
1. Database manipulation is a combination of INSERT, UPDATE, DELETE and SELECT statements.
2. Static data is loaded into the database in the form INSERT statements in SQL script files. Dynamic data is loaded into the database from application programs written in programming languages such as Java, C, C++, C# and so on. In this lab exercise, we will focus only on loading static data through SQL scripts.
3. Detailed documentation on SQL DML can be found here: <http://dev.mysql.com/doc/refman/5.1/en/sql-syntax-data-manipulation.html>
4. "INSERT" scripts: Create ONE large script file called insert_dbname.sql (e.g., insert_companydb.sql). This script file should contain:
 - a. INSERT statements to insert all the rows into the individual tables. Values for all the attributes should be assigned (but note (b) below)
 - b. The INSERT statement should assign NULL to all the foreign key attributes to start with.
5. "UPDATE" scripts: Create ONE large script file called update_dbname.sql (e.g., update_companydb.sql). This script file should contain:
 - a. All the UPDATE statements to update the foreign key attribute of all the relevant rows of all the relevant tables with appropriate values.

Note: This strategy of two separate scripts for INSERT and UPDATE is to handle any cyclic table dependencies and can be used only when you allow NULL values for FKs.

Referential Integrity Constraints for COMPANY database

Figure 5.7

Referential integrity constraints displayed on the COMPANY relational database schema.



Populated database state for COMPANY

Figure 5.6
One possible database state for the COMPANY relational database schema.

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse