Introduction to SQL

Introduction to SQL

- A standard language used in most DBMS.

- Both as a DDL and DML language.
  - **DDL** (Data Definition Language): define the schema of the database.
  - **DML** (Data Manipulation Language): provides commands to manipulate the database (query, insert, update, delete).

SQL Continued

- *Based* on relational algebra, but not entirely identical.
  - Relations ⇔ Tables
  - Tuples ⇔ Rows
  - Attributes ⇔ Columns

Basic DDL Commands in SQL

- **CREATE**: to define new tables (to define relation schemas)
- **DROP**: to delete table definitions (to delete relation schemas)
- **ALTER**: to change the definitions of existing tables (to change relation schema)
- Other features as DDL
  - Specify referential integrity constraints (FKs)
  - Specify user-defined attributes constraints

Basic DML Commands in SQL

- **INSERT**: to add new rows to table
- **UPDATE**: to change the “state” (the value) of rows.
- **DELETE**: to remove rows
- **SELECT**: a query command that uses relation algebra *like* expressions

- Various options available to handle the enforcement/violation of integrity constraints

Platforms of This Course

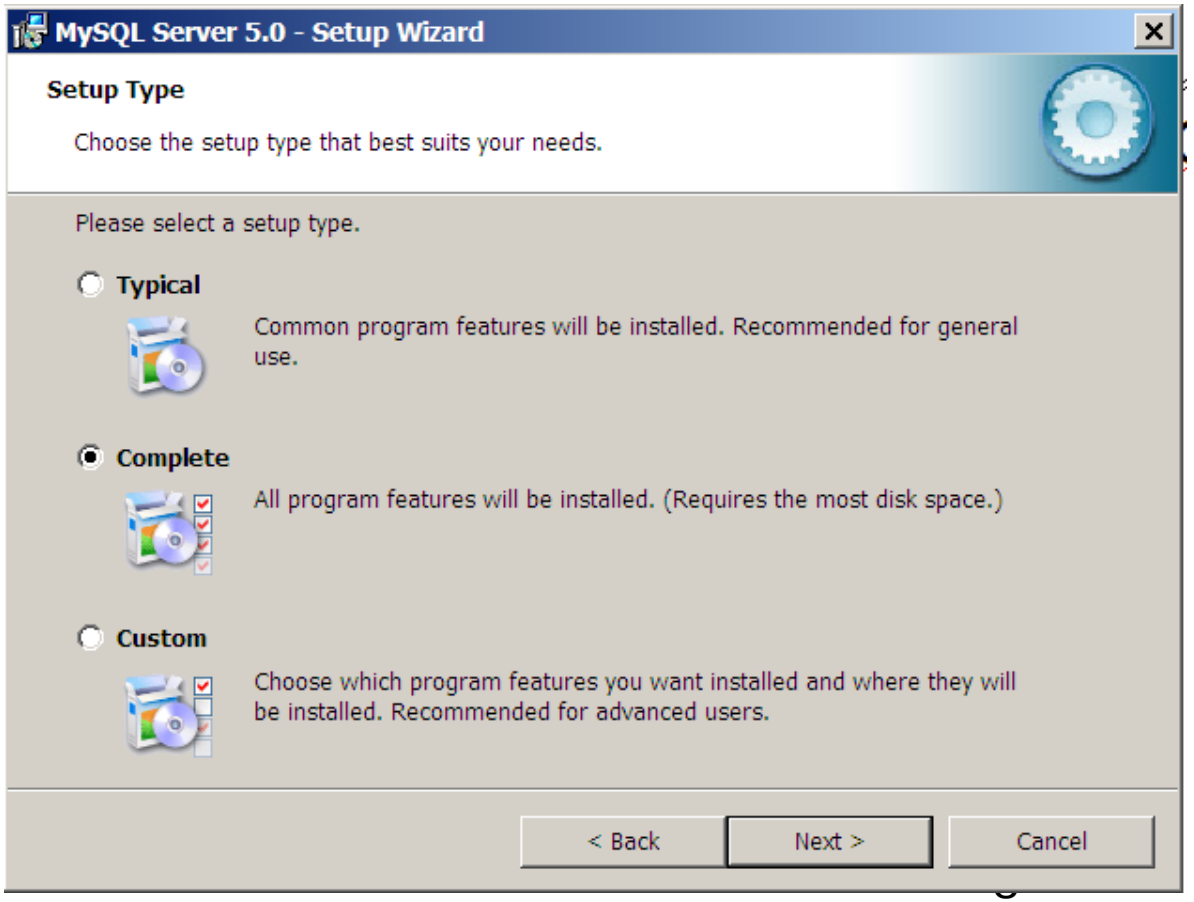
- Platform 1: MySQL
  - Open source, free software
  - Available on Windows and Linux.
  - Easily installed on your own PC.
- Platform 2: Oracle 10g Enterprise Edition
  - Available through IT&E labs
  - Proprietary, popular DBMS
  - Please see http://labs.ite.gmu.edu/reference/faq_oracle.htm for details.

MySQL

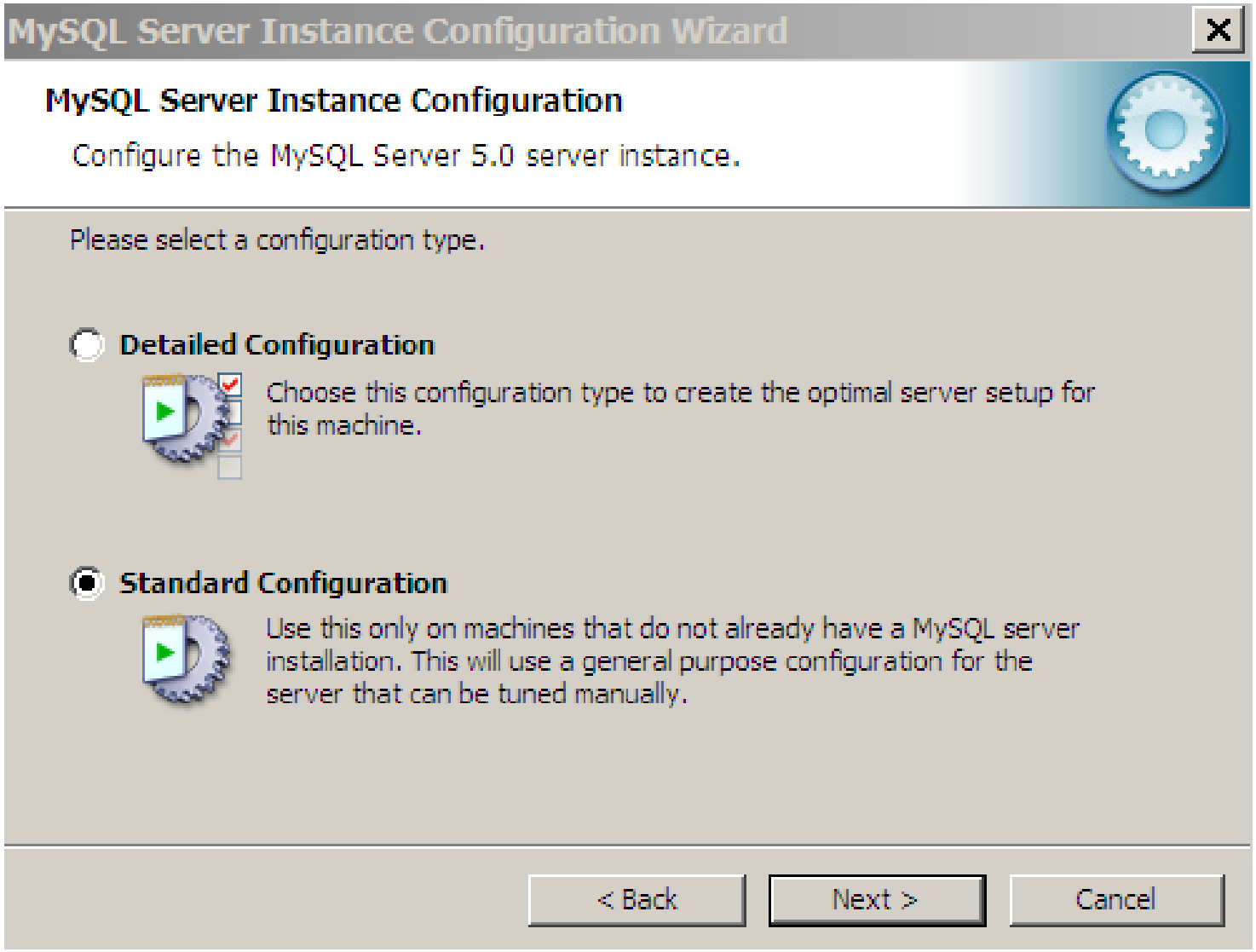
- Open source and free
- Generally not as powerful as Oracle
- Still, it is an industrial strength package.
  - Users include Amazon, NASA, Google, Yahoo …
- A commercial edition is also available (MySQL Enterprise) --- You are paying for the services.

Installation on Windows

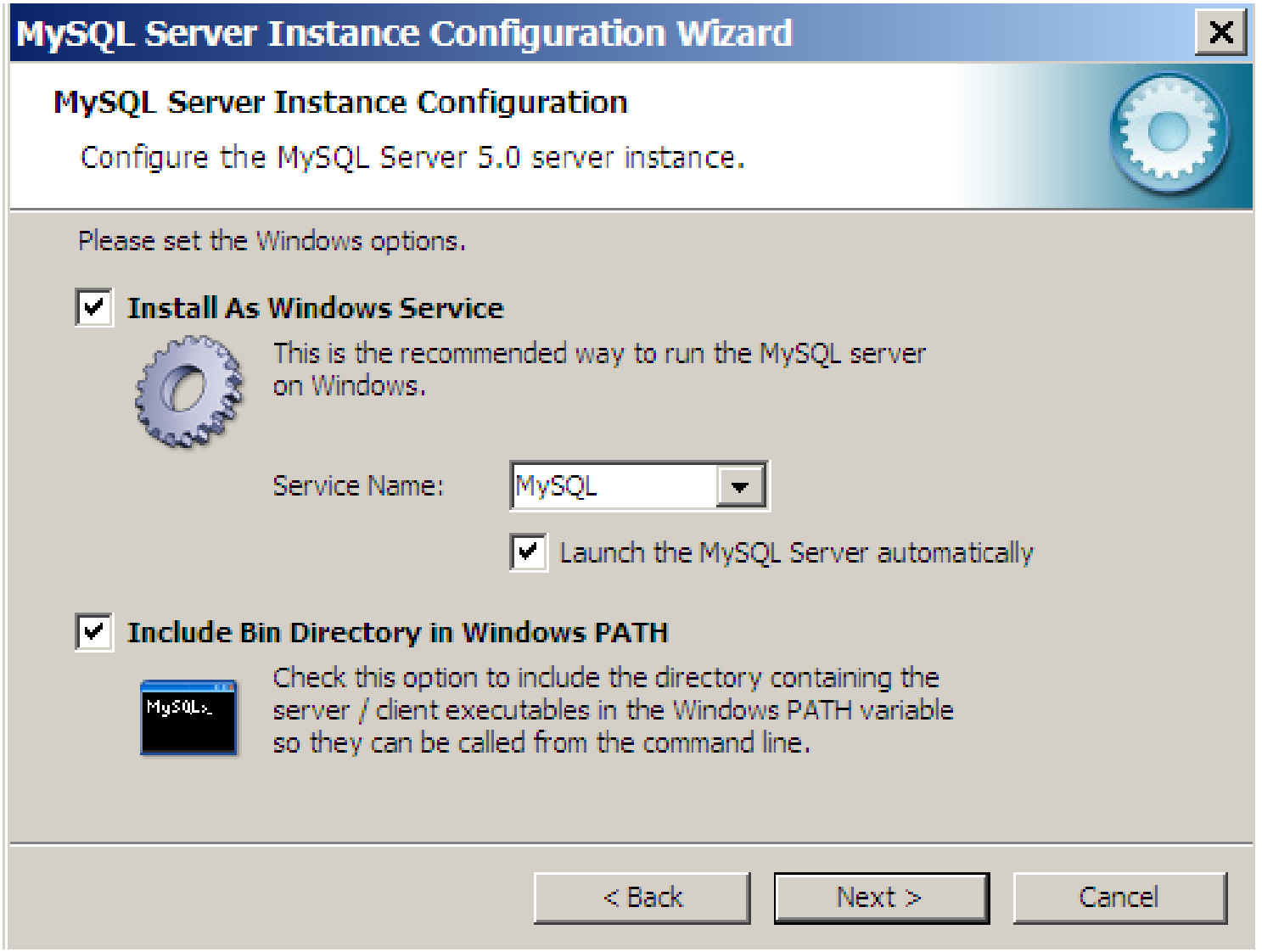
- Download the Essential Version of MySQL 5.0 from mysql.com
- Click on the .exe file to start installation.
- In Setup Type, Choose “Complete”



Use Standard Configuration



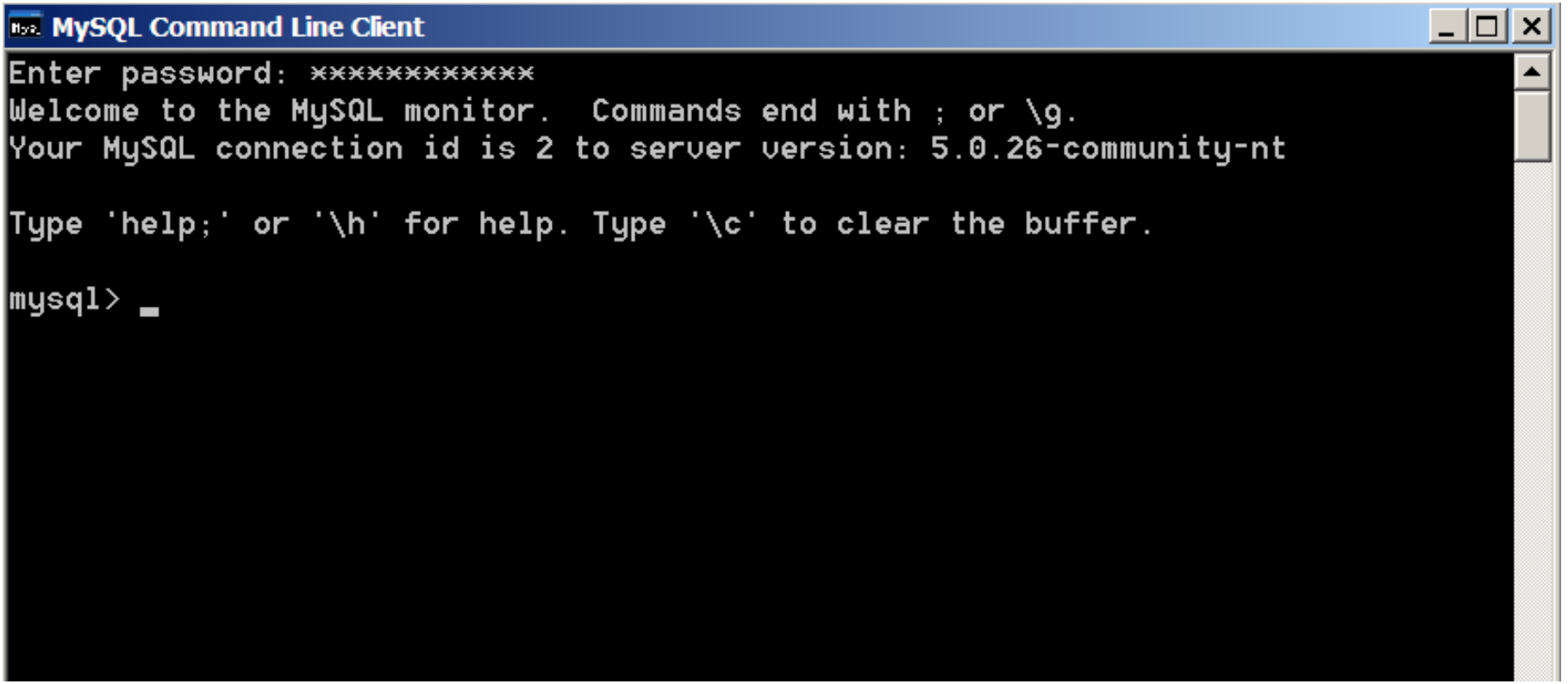
Install As Windows Service

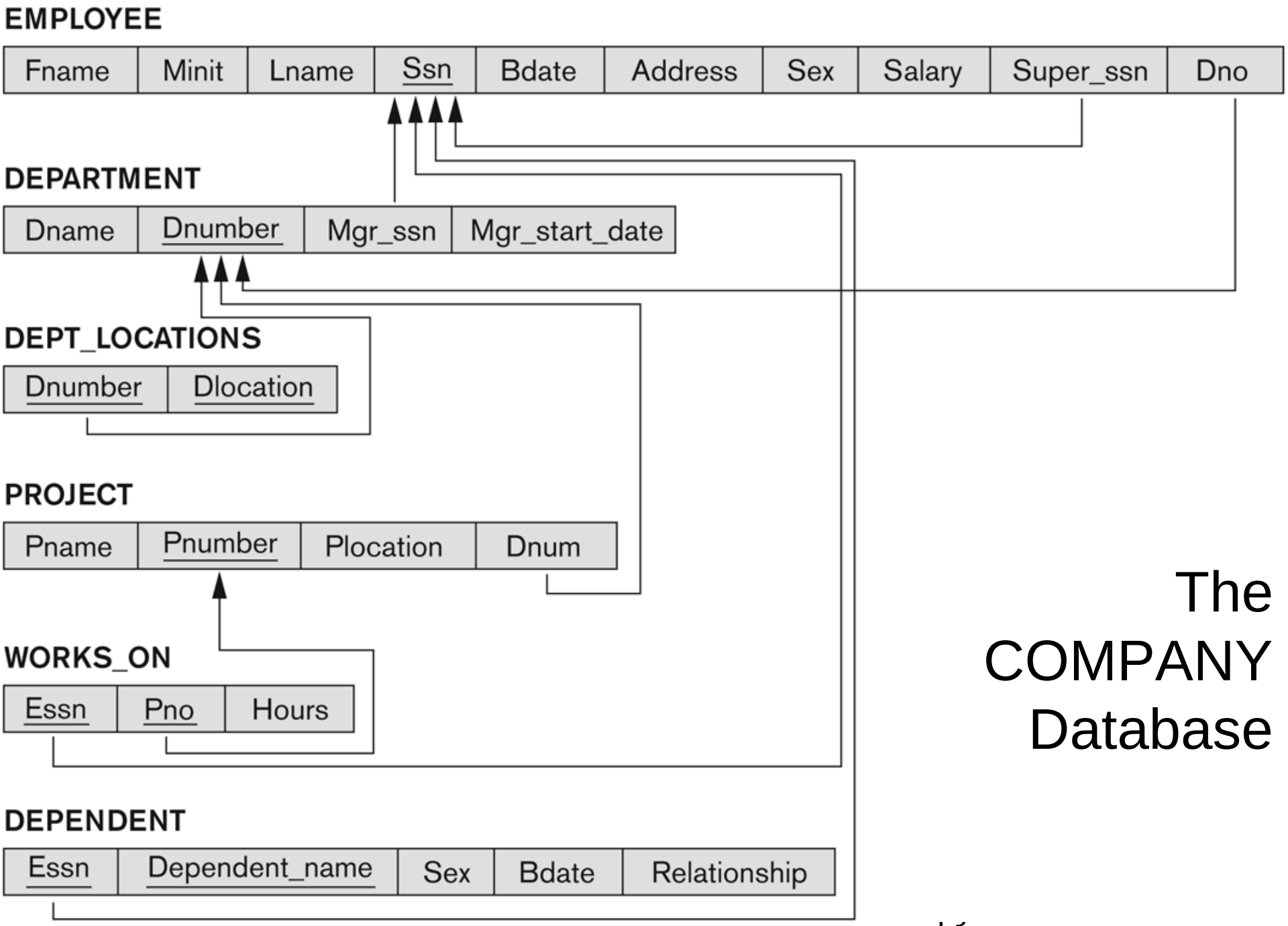


Set Root Password

Launch MySQL

- Use the Start menu to launch the “MySQL Command Line Client”
- Enter the root password





The COMPANY Database

Create the COMPANY Database

- To create

```
create datatbase COMPANY;
```

- To use (or switch to) the database

```
use COMPANY;
```

- Subsequent commands will operate on the COMPANY database by default.

CREATE TABLE

```
CREATE TABLE DEPARTMENT (
Dname       VARCHAR(10) NOT NULL,
Dnumber     INTEGER  Default 0,
Mgr_ssn     CHAR(9),
Mgr_Sartdate CHAR(9),
PRIMARY KEY  (Dnumber),
UNIQUE      (Dname),
FOREIGN KEY  (Mgr_ssn)
    REFERENCES EMPLOYEE (Ssn));
```

- The “UNIQUE” clause specifies secondary keys.
- EMPLOYE)has to be created first for the FK Mgr_ssn to refer to it.

Additional Data Types

- **DATE:**
  - Made up of year-month-day in the format yyyy-mm-dd
- **TIME:**
  - Made up of hour:minute:second in the format hh:mm:ss
- **TIMESTAMP:**
  - Has both DATE and TIME components
- **Decimal (*i*,*j*):**
  - *i*: total number of digits
  - *j*: the number of digits after the decimal point
- Others: Boolean, Float, Double Precision

Adding the Dno FK to EMPLOYEE

- If "`create table` EMPLOYEE" is issued first, we cannot specify Dno as a FK in that `create` command.
- An ALTER command must be used to change the schema of EMPLOYEE, after the "`create table` DEPARTMENT," to add a FK.

```
alter table EMPLOYEE
 add constraint
  foreign key (Dno)
     references DEPARTMENT (Dnumber);
```

The Check Clause

- Used to specify user-defined constraints
- Assume that dept. numbers are from 0 to 99.

```
create table DEPARTMENT (
  …
  Dnumber INTEGER  Default 0
      check (Dnumber>=0 AND Dumber<=99),
  …);
```

- “Check” can also be a clause of the entire table.

```
 create table DEPARTMENT (
  …
  Dept_create_date date,
  Mgr_start_date   date,
  check (Dept_create_date <= Mgr_start_date)
 );
```

Add Columns to Existing Tables

- To add spouse SSN (S_ssn) to EMPLOYEE

```
alter table EMPLOYEE add column S_ssn char(9);
```

  - The new attribute will have NULLs in all the tuples of the relation right after the command is executed

- Alternatively, we can set a default value.

```
alter table EMPLOYEE add column S_ssn char(9)
    default "000000000";
```

Delete Columns from Existing Tables

- To delete column S_ssn

```
alter table EMPLOYEE drop column S_ssn;
```

- **Reminder:** changing relation schemas typically indicates ill-executed design phase of the database.

Referential Integrity Options

- **Causes** of referential integrity violation for a foreign key FK (consider the `Mgr_ssn` of `DEPARTMENT`).
  - **On Delete**: when deleting the foreign tuple
    - What to do when deleting the manager tuple in `EMPLOYEE` ?
  - **On Update:** when updating the foreign tuple
    - What to do when updating/changing the SSN of the manager tuple in `EMPLOYEE` is changed ?
- **Actions** when the above two causes occur.
  - **Set Null**: the `Mgr_ssn` is set to null.
  - **Set Default**: the `Mgr_ssn` is set to the default value.
  - **Cascade**: the `Mgr_ssn` is updated accordingly
    - If the manager is deleted, the department is also deleted.

The Mgr_ssn Example

```
CREATE TABLE DEPARTMENT (
    …
    Mgr_ssn    CHAR(9),
    …
    FOREIGN KEY (Mgr_ssn)
       REFERENCES EMPLOYEE (Ssn)
        ON DELETE  ???
        ON UPDATE  ???
  );
```

Another Example

```
Create table EMP(
  …
  ESSN          CHAR(9),
  DNO           INTEGER  DEFAULT 1,
  SUPERSSN      CHAR(9),
  PRIMARY KEY (ESSN),
  FOREIGN KEY (DNO) REFERENCES DEPT
          ON DELETE SET DEFAULT
          ON UPDATE  CASCADE,
  FOREIGN KEY (SUPERSSN) REFERENCES EMP
          ON DELETE SET NULL
          ON UPDATE CASCADE);
```

Another Example

```
Create table EMP(
  …
  ESSN          CHAR(9),
  DNO           INTEGER  DEFAULT 1,
  SUPERSSN      CHAR(9),
  PRIMARY KEY (ESSN),
  FOREIGN KEY (DNO) REFERENCES DEPT
          ON DELETE SET DEFAULT
          ON UPDATE  CASCADE,
  FOREIGN KEY (SUPERSSN) REFERENCES EMP
          ON DELETE SET NULL
          ON UPDATE CASCADE);
```

Miscellaneous Commands

- `show databases;`
  - Show all the databases on the server
- `show tables;`
  - Show all the tables of the present database
- `show columns from table EMPLOYEE;`
- `drop table` *`t_name`*`;`
  - Delete the entire table *t_name*
- `drop database` *`db_name`*`;`
  - Delete the entire database *db_name*
- `load data infile` *`f_name`* `into table` *`t_name`*`;`
  - To be discussed with the next homework.