Structured Query Language (SQL)

Basic Statements

Connecting to Oracle

SQL

- Four types of statements
  - Data Definition Language (DDL)
    - Defining the database's schema
    - Defining the contents of the data dictionary (DD)
  - Data Manipulation Language (DML)
    - Select, Insert, Delete, Update
  - Database Control Language (DCL)
    - Grant and revoke
  - Vendors additions
    - Stored procedures and triggers

SQL

- Creating the structures/schema of the database
  - Create … command
    - Is used to create different structures within the relational database
  - Alter … command
    - Is used to modify the existing structures of the relational database
  - Drop … command
    - Is used to discard the existing structures within the relational database

SQL

- Relational Database Structures
  - Create Table …
    - Constraints
    - Referential Rules
    - Defaults
  - Create User defined data types
  - Create Index …
  - Create View …
  - Create Stored Procedure …
  - Create Trigger …
  - Etc.

Database Integrity

- Data stored in a database must satisfy some correctness and quality rules.
- These correspond to the business rule defined by the organization
- These rules are called “Integrity” rules or “Constraints”
- Constraints are defined:
  - Single table constraints
  - Two table constraints (Referential Integrity or RI rules)
  - Multi-table constraints

Constraints

- Single table constraints
  - Identity Constraints
    - PRIMARY KEY constraint
    - UNIQUE constraint
  - Non-identity Constraints
    - DATA TYPE constraint
    - DEFAULT definition
    - CHECK constraint (including user defined checks)
    - NOT NULL constraint
- Constraints Across tables
  - Referential Integrity Constraints
    - PRIMARY KEY/FOREIGN KEY constraint
- General Assertions
  - Can apply to one or more tables
  - Oracle does not support this
- User-defined Constraints
  - Triggers
  - Stored Procedure

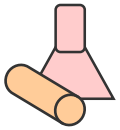
Create Table Command

```
Create Table Employee
    ( Fname          varchar(15)        not null,
      Minit          char(1),
      Lname          varchar(15)        not null,
      SSN            char(11)           not null,
      Bdate          date,
      Address        varchar(30),
      Sex            char(1),
      Salary         Number(10,2),
      SuperSSN              char(11)          not null,
      DNO            Number(2)                Default 1,
Primary Key (SSN),
Foreign Key (SuperSSN) References Employee(SSN)
On Delete Restrict  On Update Restrict,
Foreign Key (DNO) References Department(Dnumber)
On Delete Set Default  On Update Cascade,
CHECK (Sex IN ('F', 'M')));
```

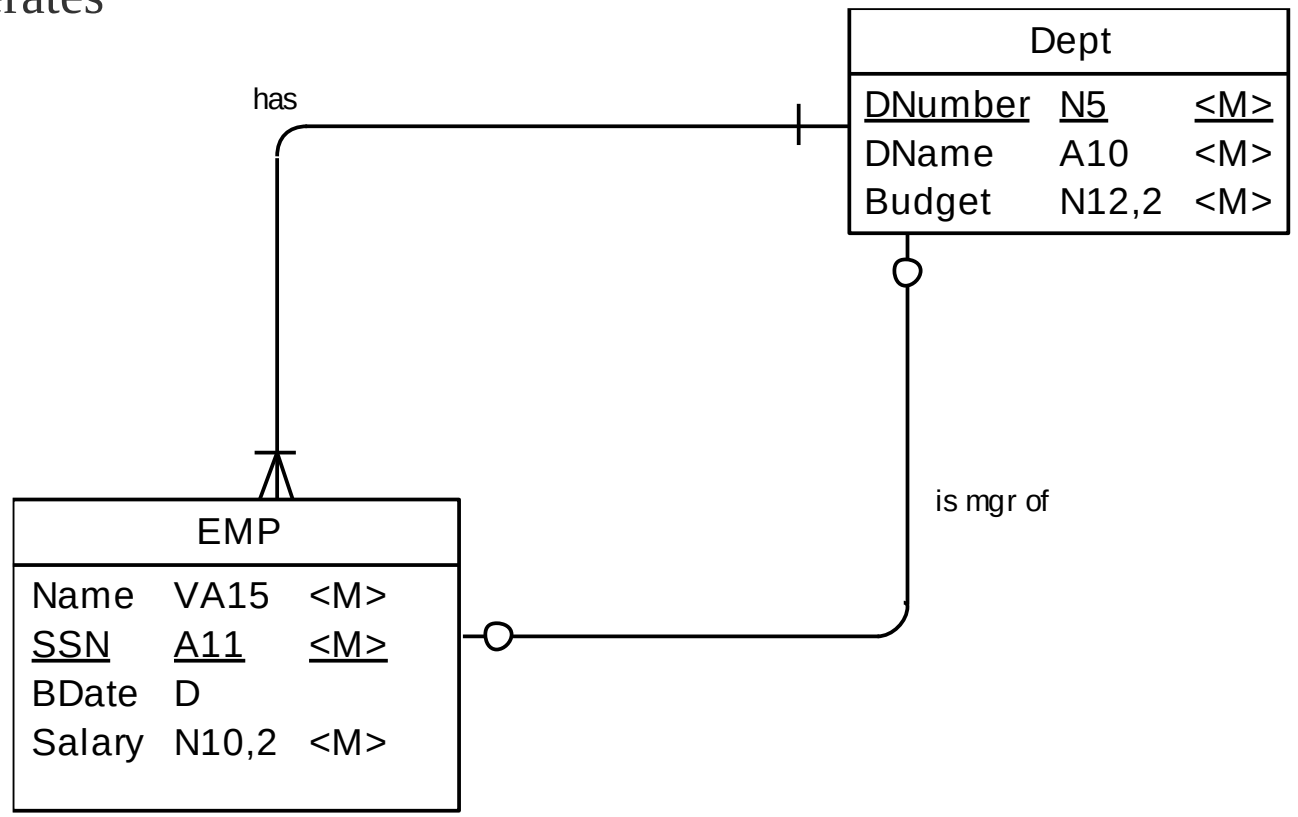
Create Table Command

```
Create Table Employee
   ( Fname          varchar(15)        not null,
     Minit          char(1),
     Lname          varchar(15)        not null,
     SSN            char(11)           not null,
     Bdate          date,
     Address        varchar(30),
     Sex            char(1),
     Salary         Number(10,2),
     SuperSSN              char(11)          not null,
     DNO            Number(2)                Default 1,
Constraint EMPPK Primary Key (SSN),
Constraint EMPSUPERFK  Foreign Key (SuperSSN) References
    Employee(SSN)
On Delete Restrict  On Update Restrict,
Constraint EMPDEPFK Foreign Key (DNO) References
    Department(Dnumber)
On Delete Set Default  On Update Cascade,
Constraint CK_SEX  CHECK (Sex IN ('F', 'M')));
```

Create Table Command



- Write the DDL statements for the tables that the following ER diagram generates



Create Table Command

```
/*==============================================================*/
/*==============================================================*/
/* Table: DEPT                                                  */
/*==============================================================*/
create table DEPT
(
   DNUMBER              NUMBER(5)           not null,
   SSN                  CHAR(11),
   DNAME                CHAR(10)            not null,
   BUDGET               NUMBER(12,2)        not null,
   constraint PK_DEPT primary key (DNUMBER)
);

/*==============================================================*/
/* Table: EMP                                                   */
/*==============================================================*/
create table EMP
(
   NAME                 VARCHAR2(15)        not null,
   SSN                  CHAR(11)            not null,
   DNUMBER              NUMBER(5),
   DEP_DNUMBER          NUMBER(5)           not null,
   BDATE                DATE,
   SALARY               NUMBER(10,2)        not null,
   constraint PK_EMP primary key (SSN)
);
```

Create Table Command

```
/*==============================================================*/
/* Add Foreign key after all tables have been created           */
/*==============================================================*/

alter table DEPT
   add constraint FK_DEPT_IS_MGR_OF_EMP foreign key (SSN)
      references EMP (SSN);

alter table EMP
   add constraint FK_EMP_HAS_DEPT foreign key (DEP_DNUMBER)
      references DEPT (DNUMBER);

alter table EMP
   add constraint FK_EMP_IS_MGR_OF_DEPT foreign key (DNUMBER)
      references DEPT (DNUMBER);
```

Create Table Command

```
Create Table Emp_Proj_Hours
    ( SSN       Number(2)                   not null,
      PID       Number(2)                   not null,
      Hours     Number(2)                   not null,
Primary Key (SSN, PID),
Foreign Key (SSN) References Employee (SSN)
Foreign Key (PID) References Proj(PID));
```

Dropping Table

- Dropping Tables:
  - Oracle by default does not allow dropping a parent table when there are children tables (or other objects depending on it).
  - This gives Oracle a restricted drop action
  - Example:
    - Emp (Eno, Ename, Sal)
    - Child (Cname, Eno, age)
    - Where Child.Eno is a fkey pointing to Emp.Eno as pkey.
- In this case,
  - “Drop table Emp” will fails since that will leave orphan children rows in the Child table.
  - To force the drop, we will use,
    - Drop table Emp cascade constraints;

Indices

- Indices
  - Index is a structure that allow direct access to the rows of a table
  - It provides for query performance improvement
  - It can also slow down modification statement's performance
- Examples:

```
Create Index Name_index on Employee(Lname);
Create Index Lname_Index on Employee(Lname Asc);
Create Index Names on Employee(Lname Asc, Fname Desc);
Create Unique Index empID_index on Employee(SSN);
```

  - Typically, one should index the primary key column(s) and the foreign key column(s) to improve the join performance

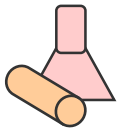
Hands-on

- The Bank Database Table
  - A small bank has branches in which customers open up accounts and/or loans
  - Branch names are unique (Bname). For each branch, we keep track of the city where the branch is (Bcity) and the assets of the branch.
  - Each customer has a unique name (Cname). For customers, we track the street address (street) and the city (Ccity).
  - A customer may have one or more account or loan opened in branches in the bank.
  - Accounts have a unique account number (A#) and the balance (bal).
  - Loans have a unique loan number (L#) and the amount of the loan (amt).
  - Create the ERD for this bank and generate the DDL statements to create the tables.

Business Rules

- Business rules in the database reflect the constraints that organizations have on data items of a database.
- Examples of business rules:
  - Every employee must work for a department
  - Every department must have a department manager
  - Age of employees must be a positive number
  - In Oracle, a rule is assigned to a column by a “Check Constraint”
  - SQL Server can use check constraint and also has rule as a separate construct

Oracle Rules



- Create a table called Min_cities.
- This table has two columns
  - Column city_name is varchar(20) and can only accept city names (MPLS, STP, STC, MNKTO)
  - Column population Number (8)
- Show how this rule works by trying to insert an invalid city into the database
- Remove the rule from the column it is attached to

Oracle Rules

- Answer:

```
SQL> create table Mn_cities (
  2     city_name varchar(20),
  3     population number(8),
  4  constraint nameCT check (city_name IN ('MPLS', 'STP',
'STC', 'MNKTO')));

Table created.

SQL> insert into Mn_cities values ('MPLS', 2000000);

1 row created.

SQL> insert into Mn_cities values ('Minneapolis', 2000000);
insert into Mn_cities values ('Minneapolis', 2000000)
*
ERROR at line 1:
ORA-02290: check constraint (SKR.NAMECT) violated


SQL> alter table Mn_cities drop constraint nameCT;

Table altered.
```

Database Control

- Access to the database is controlled by the use of Grant/Revoke statements
- Syntax:

  Grant DBA|select|update|insert|delete
  On {table_name[.column][,]}$^{1..*}$
  To user_name|group_name
  [with Grant Option];

- Examples:

  Grant select on employee to public;
  Revoke update on employee from Saeed;
  Grant DBA to Saeed with grant option;

Database Control

- Example:
  - Supposed Sam creates table Emp
    - Therefore Sam owns table Emp.
  - If Sam wants Joe to be able to select, insert, delete, update the table, Sam issues the following:

    Grant select, insert, update, delete on Emp to Joe;
  - Joe then can use the table:

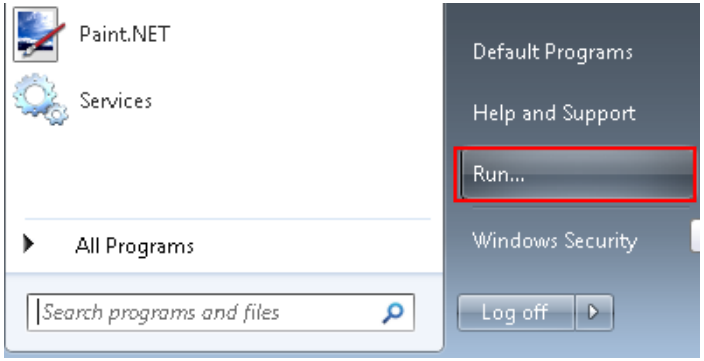
    Select * from sam.Emp;

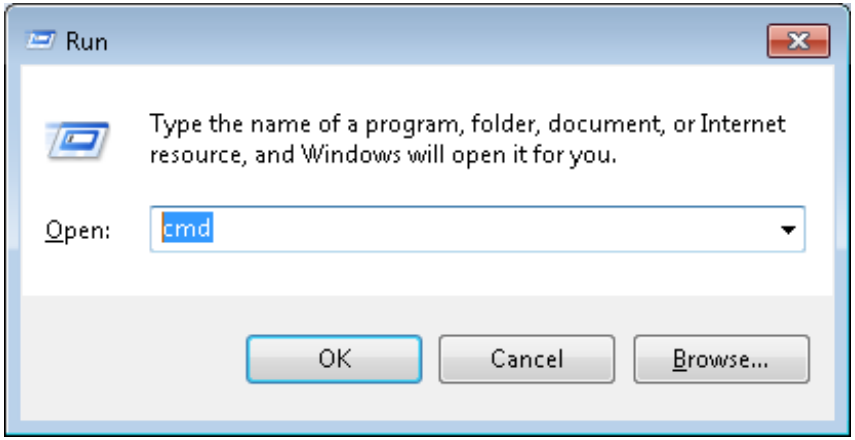
    Insert into sam.Emp values (…..);

Working with Oracle XE

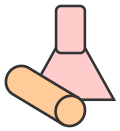
Connecting to Oracle Database

- Using Command Line in Windows
  - From the **Start Menu,** select **Run,** then type **cmd** in the run window and click "OK"



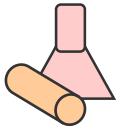


Connecting to Oracle Database



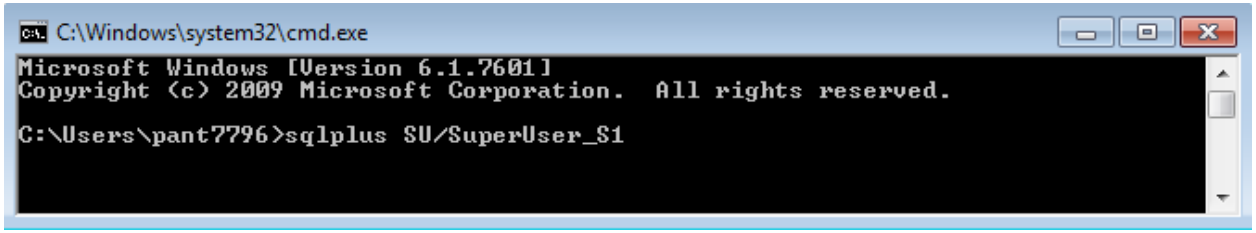
- Customizing the Command Windows
  - Pin the program to the taskbar for quick access, if you like.
  - Click on c:\icon in the upper left corner of the window and choose “Defaults”
    - Follow the steps specified in class to customize the CMD window

Connecting to Oracle Database



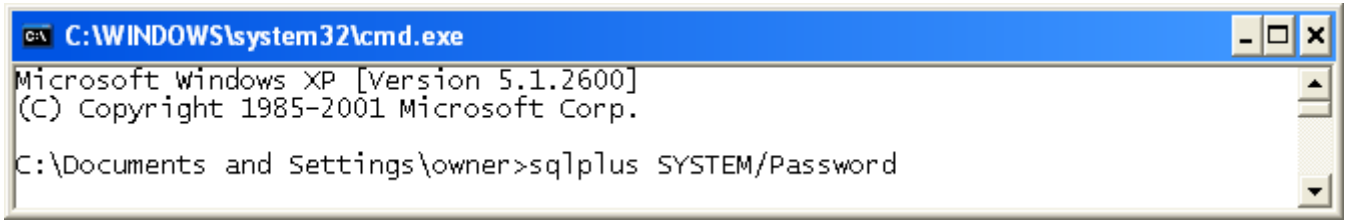
- For the desktop computers in class type:

  sqlplus SU/SuperUser_S1

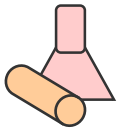


- For your own laptop type:

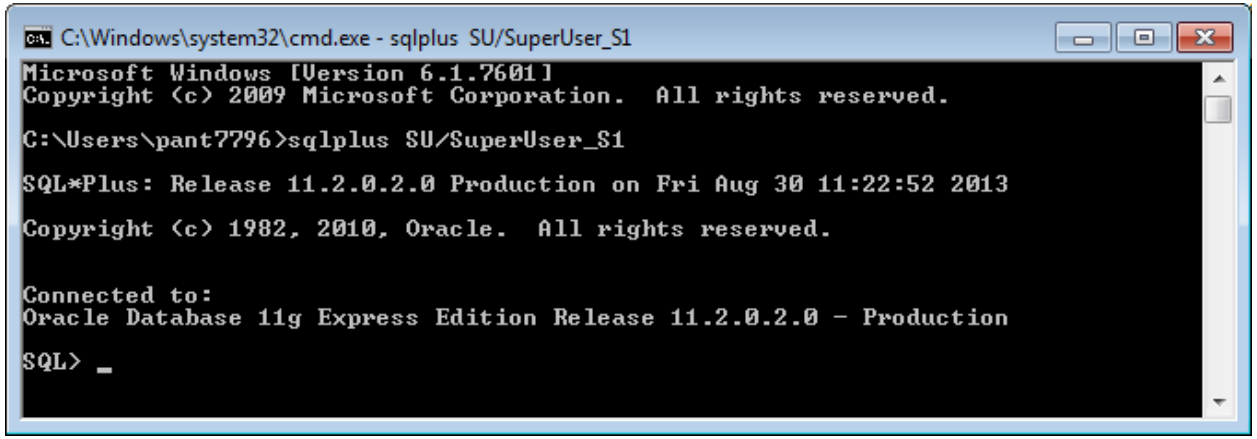
  sqlplus System/System's Password



Connecting to Oracle Database

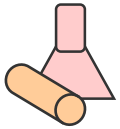


- The **SQL>** prompt will then appear



- Important Note:
  - You ***must not*** do any of your work logged in as SuperUser (SU) or SYSTEM
  - Follow the steps given in class to create a user you will use to do your work

Connecting to Oracle Database

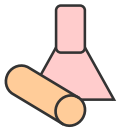


- Once you created the user in the previous step, type “Exit” at the command prompt

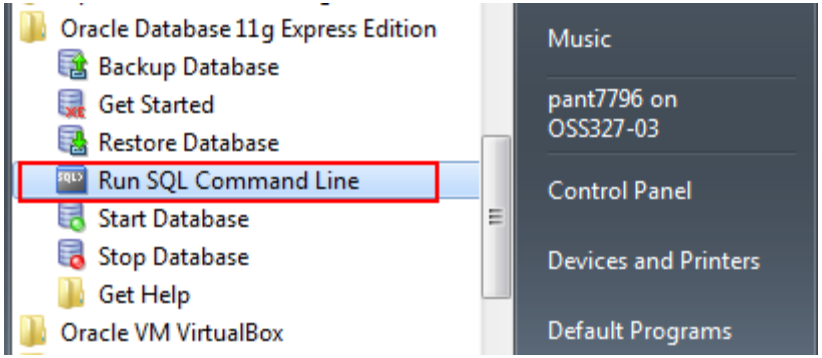
  SQL> Exit;

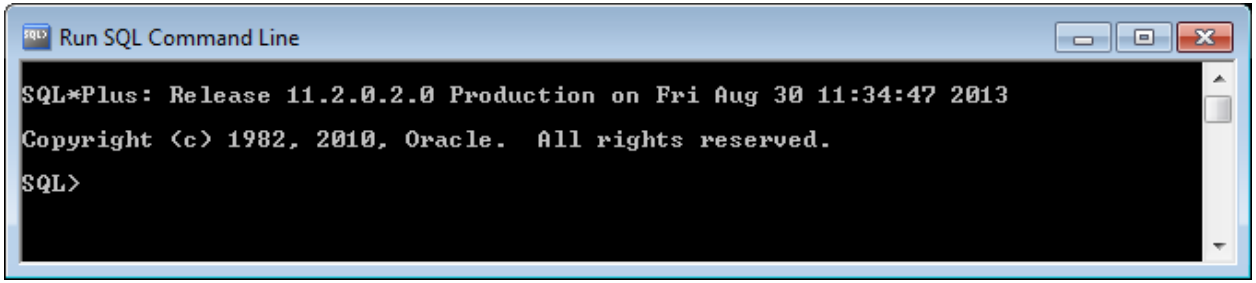
- Then close the command window

Connecting to Oracle Database

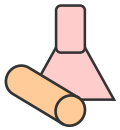


- Connecting to Oracle XE using the user you just created
  - Select **Start -> All Programs**
  - Under **Oracle Database 11g Express Edition,** select **Run SQL Command Line**
    - **This runs the interface but does not connect the user**





Connecting to Oracle Database

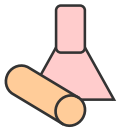


- Connecting to Oracle XE using the user you just created
  - The previous step runs the interface but does not connect the user
  - To connect, use the user information you just created and type:

    SQL> connect username/password
  - When you get the SQL prompt back you are connected.
  - To exit, type in;

    SQL> Exit;
  - Close window

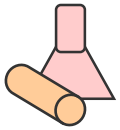
Connecting to Oracle Database



- Connecting to Oracle XE using the user you just created
  - As an alternative to running the SQL command line utility and then connecting to the server, you can connect in one step.
  - Open the Windows command window you customized.
  - At the prompt, type

    C:\sqlplus username/password
  - Once you get the SQL> prompt, you are connected to the server as this user.

Setting up the Database we use



- Get the files needed for the bank database
  - Download O_bankDDL.sql from the course website and save it to your working directory
  - Download O_bankdata.sql from the course website and save it to your working directory
- Connect to Oracle XE
  - Creating the tables for the bank database
    - SQL >**Start O_bankDDL.sql;**
    - Check for errors
  - Loading information into the bank database tables
    - SQL >**Start O_bankdata.sql;**
    - Check for errors

Setting up the Database we use

- Scripting in Oracle
  - Open the file L3_Script_file.sql from your working directory in Notepad++ (or any text editor)
  - Edit the file and replace U:\XYZ with your working directory

  - **Save** the modified file as ASCII text. For example, **U:\XYZ\MySource.sql**

Setting up the Database we use

- Testing your script
  - Connect to Oracle XE database if you are not connected already
  - At SQL prompt type the following:

    SQL > start **U:\XYZ\MySource.sql**
  - Once you get the SQL prompt back, type

    SQL> Exit;
  - Check the file: **U:\XYZ\MyResult.txt**
  - When you do any SQL homework, you will have two files
    - One with the extension of .SQL – this is the source file (or source script)
    - One with the extension of .TXT – this is the result file