BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE
PILANI, RAJASTHAN, 333031

DATA WAREHOUSING
[SS G515]
II Sem 2011-12
LAB #1
GETTING STARTED WITH SQL

1. To list all the objects available in your database

- Execute the following SQL query

```
SQL> Select * from tab;
```

and it will list all the tables, views, synonyms etc that are available with you.

2. Editing the last executed query

- It is possible that you may have written an SQL that did not work. In that case it is not necessary for you to rewrite the same again. SQL allows you to edit the last executed query. Just execute the command edit and

SQL> EDIT (can write shorter version “ed” also)

SQL will call the default editor defined over there (mostly Notepad). After editing the query, save it and exit. SQL will display the whole query again on the screen. Now the next thing is to execute it, so type the following at the prompt

SQL> / (slash) or **SQL> run**

3. Saving the last executed query in a file for future use

- Write the following command at the SQL prompt

```
SQL> SAVE filename
```

The above command shall be saving the contents of the buffer i.e. the last executed query into the file whose name is specified in the default directory (mostly C:\orant\bin) and with the extension .sql. Alternatively you can specify the full path also as follows:

```
SQL> save d:\users\my_sql\sql1
```

4. Executing an SQL query saved in a file

- There are various ways of doing the same. We shall be covering two here:

First way : Type the following command at the SQL prompt: **SQL> @filename**

The above command will do two things. It will bring the contents of the file in the buffer and then execute them. Obviously, the file has to be in the default path or you have to specify the full path.

Second way : Firstly get the contents of the file into the buffer by executing the following command

SQL> GET filename (useful when you want to modify the SQL before executing)

and then execute it either by using slash or the run command.

5. Doing copy and paste

Inside SQL window you can perform copy and paste also as normally you do in other windows application. Use **<ctrl>-c** for copying and **<ctrl>-v** for pasting. We shall not be exploring on this as most of you must be knowing it already. The copy and paste can be done to and from other applications also.

6. Data types in Oracle

- Although there are lots of data types available in Oracle 9i but for our purpose we shall be covering only the following:

Datatype	Format	Explanation
Number	Number(m[,n])	Here, m is the total number of digits in the number and n signifies number of decimal digits. A square bracket means the contents inside it are optional and in that case the number become an integer.
Character	Char(n)	This is for fixed length strings and n signifies the maximum number of characters.
Varchar2	Varchar2(n)	For variable length strings and n signifies the maximum number of characters.
Date	Date	For storing date values

Other data types shall be introduced as and when required.

7. Creating your first object (a table without any integrity constraints)

- To create a table we use the "Create table …" DDL.
- Execute the following DDL at the SQL prompt

SQL> CREATE TABLE STUDENT(IDNO CHAR(11), NAME VARCHAR2(30), DOB DATE, CGPA NUMBER(4,2), AGE NUMBER(2));

If everything you typed is as it is then the DBMS will create the table and display the message "Table created" else it will give an error message. In that case use the edit command and do the necessary changes and re-execute the query.

8. Inserting records in the table

- To insert records in a table we shall be using the "Insert into table…" DDL. There are two forms of it.

First form : Here we need to specify values for all the columns and in the same order as they were specified at the time of table creation. For a particular column that can take null values we can specify it's value as NULL also.

SQL> INSERT INTO student VALUES('1997B2A5563','K Ramesh','21-Jun-75',7.86,27);

The above Insert statement will create a new record in the student table and insert these values in the corresponding columns. Note that all the char, varchar2 and date literals are specified in single quotes and also the format of date value is "dd-mon-yy" which is the default format. Using the above SQL statement you can insert as many records as you want.

Second Form : In the second form we can use column names in the SQL statement and this gives us a lot of flexibility. Firstly, we can omit any column and secondly, we don't need to maintain any order.

SQL> INSERT INTO student(name,idno,age) VALUES('R Suresh','1998A7PS003',25);

The above Insert statement will create a new record and inserts the specified values into the corresponding columns. It will insert NULL values in those columns that are not specified here except… Hence only those columns which can take NULL values should only be omitted. Can you write the first stmt in the second form?

Note that specifying a column as of type char doesn't guarantee that it will accept exactly that much number of characters only. In those cases where you will specify lesser number it will simply pad blank spaces.

9. Using variables in SQL queries

- You can also write the above Insert statements as follows:

SQL> INSERT INTO student VALUES('&id','&name','01-Jan-75',&cgpa,27);

Here id, name and cgpa shall be taken as variables and there values will be asked by the user at runtime (dob and age are static). Once the user specifies all the values, the DBMS will replace the variables by the actual values and then execute the same. Using this you can save some of your typing effort.

10. Committing or rolling back the work done by you till now

- If you want to save the work done by you till now (applicable only for insert, update or delete operations) then execute the following command:

  SQL> COMMIT; (normal exit from Oracle (by using **"EXIT"** command) also commits the changes)
- And if don't want to make the changes permanent then execute the following command:

  SQL> ROLLBACK; (abnormal exit like power failure, externally closing the window etc. rollback the changes)

Things you did today:

1.Getting familiar with sql environment

2. Primitives data types *3 Create a table* *4. Insert values in a table*

5. Commit and rollback.

Exercise :

1. Create the following tables for the Suppliers-and-Parts database in and insert the given tuples.

Supplier

SNO	SNAME	STATUS	CITY
S1	SMITH	20	LONDON
S2	JONES	10	PARRIS
S3	BLAKE	30	PARRIS
S4	CLARK	20	LONDON
S5	ADAMS	30	ATHENS

Part

PNO	PNAME	COLOR	WEIGHT	CITY
P1	NUT	RED	12	LONDON
P2	BOLT	GREEN	17	PARIS
P3	SCREW	BLUE	17	ROME
P4	SCREW	RED	14	LONDON
P5	CAM	BLUE	12	PARIS
P6	COG	RED	19	LONDON

Shipment

SNO	PNO	QTY
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

2. Create the tables mentioned below for the Suppliers-and-Parts-and-Projects database with the necessary modifications of the tables given above using the Data Definition Language in SQL.

Supplier

SNO	SNAME	STATUS	CITY
S1	SMITH	20	LONDON
S2	JONES	10	PARRIS
S3	BLAKE	30	PARRIS
S4	CLARK	20	LONDON
S5	ADAMS	30	ATHENS

Part

PNO	PNAME	COLOR	WEIGHT	CITY
P1	NUT	RED	12	LONDON
P2	BOLT	GREEN	17	PARIS
P3	SCREW	BLUE	17	ROME
P4	SCREW	RED	14	LONDON
P5	CAM	BLUE	12	PARIS
P6	COG	RED	19	LONDON

Project

JNO	JNAME	CITY
J1	SORTER	PARIS
J2	DISPLAY	ROME
J3	OCR	ATHENS
J4	CONSOLE	ATHENS
J5	RAID	LONDON
J6	EDS	OSLO
J7	TAPE	LONDON

SPJ

SNO	PNO	JNO	QTY
S1	P1	J1	200
S1	P1	J4	700
S2	P3	J1	400
S2	P3	J2	200
S2	P3	J3	200
S2	P3	J4	500
S2	P3	J5	600
S2	P3	J6	400
S2	P3	J7	800
S2	P5	J2	100
S3	P3	J1	200
S3	P4	J2	500
S4	P6	J3	300
S4	P6	J7	300
S5	P2	J2	200
S5	P2	J4	100
S5	P5	J5	500
S5	P5	J7	100
S5	P6	J2	200
S5	P1	J4	100
S5	P3	J4	200
S5	P4	J4	800
S5	P5	J4	400
S5	P6	J4	500

3. Write the following queries in SQL.
   a. Get the full details of the projects.
   b. Get the full details of the projects in London.
   c. Get the supplier numbers for suppliers who supply project J1.
   d. Get all shipments where the quantity is in the range 300 to 750 inclusive.
   e. Get all part-color/ part-city combinations.
   f. Get all supplier-number/ part-number/ project-number triples such that the indicated supplier, part and project are collocated.
   g. Get all supplier-number/ part-number/ project-number triples such that the indicated supplier, part and project are not collocated.
   h. Get all supplier-number/ part-number/ project-number triples such that no two of the indicated supplier, part and project are collocated.
   i. Get part numbers of parts supplied by a supplier in London.
   j. Get part numbers of parts supplied by a supplier in London to a project in London.

k. Get all pairs of city names such that a supplier in the first city supplies a project in the second city.
l. Get part numbers for parts supplied to any project by a supplier in the same city as that of that project.
m. Get project numbers for projects supplied by at least one supplier not in the same city.
n. Get all pairs of part numbers such that some part numbers supplies both the indicated parts.
o. Get the total number of projects supplied by supplied by S1.
p. Get the total quantity of part P1 supplied by supplier S1.
q. Get part numbers of parts supplied to some project in an average quantity of more than 320.
r. Get the supplier numbers for suppliers supplying at least one part supplied by at least one supplier who supplies at least one red part.
s. Get project numbers for projects supplied with part P1 in an average quantity greater than greatest quantity in which any part is supplied to project J1.
t. Get project numbers for projects supplied entirely by supplier S1.
u. Get all cities in which at least one supplier, part, or a project is located.
v. Get part numbers for parts that are supplied either by a London supplier or to a London project.
w. Get the supplier number for suppliers who supply the same part to all projects.
x. For each part being supplied to a project, get the part number, the project number, and the corresponding total quantity.
y. Get the project numbers for projects whose city in first in the alphabetic list of such cities.