

---

## C3. Simple SQL queries. Using APEX SQL workshop. Inserting, updating and deleting data

The Structured Query Language (SQL) is a declarative language designed to help management and interrogation of relational databases. It is based on relational algebra developed by F. Codd in early '70. The American National Standards Institute (ANSI) adopted SQL as a new standard in databases in 1986. One year later, the International Organization for Standardization also adopted it as an ISI standard. Nowadays, the SQL'92 revision is the standard adopted by most RDBMS vendors. The aim of this chapter is to provide an overview of using SQL in APEX for basic data management.

### 1.1. Simple SQL Queries.

The SQL language is not case sensitive and includes several language elements: clauses, statements, expressions, predicates, and queries. It consists of a two parts: data definition language (DDL) and a data manipulation language (DML) part.

The main SQL DDL instructions are the following:

- CREATE DATABASE / DROP DATABASE
- CREATE TABLE / DROP TABLE / ALTER TABLE
- CREATE INDEX / DROP INDEX / ALTER INDEX
- CREATE VIEW / DROP VIEW/ ALTER VIEW
- CREATE TRIGGER / DROP TRIGGER / ALTER TRIGGER
- ...

The main SQL DML instructions are the following:

- 
- **SELECT**
  - **INSERT**
  - **UPDATE**
  - **DELETE**
  - **UNION**
  - ...

The most common operation in SQL is the **SELECT** query, which is performed with the pure declarative **SELECT** statement. **SELECT** retrieves data from one or more tables, but can also compute some expressions by combining these data. The **SELECT** statements have no persistent effects on the database. The result is returned to the “user” (the entity that sent the query to the DB server), which has the responsibility of using data in various ways (e.g. displaying on a GUI, printing a report, computing complex calculations etc.).

The basic syntax of the **SELECT** operation that allow us to retrieve some data from relational tables is:

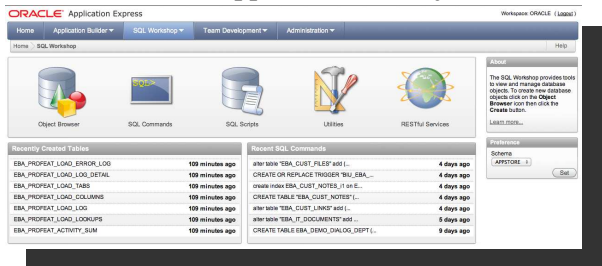
```
SELECT projection_list | *  
FROM table  
[ORDER BY list_of_fields ASC|DESC];
```

The `projection_list` is used to implement the PROJECTION relational operator and is formed by a comma-separated list of database fields (or expressions). Only these fields will be included into result. If the `*` marker is specified instead the `projection_list`, all fields from the original table will be included into result, and the order of them will be the same as in the input table.

## 1.2. Using the APEX SQL Workshop.

The SQL Workshop is an APEX module intensively used in Oracle Application Express database development. It provides functionalities for loading and unloading data from an Oracle

database, generate DDL queries, view and manipulate data and object, and even restore dropped database objects.

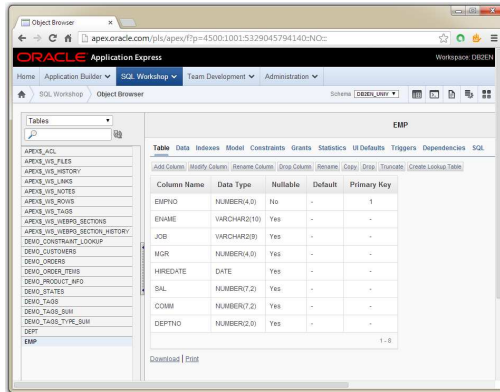



The SQL Workshop offers four components:

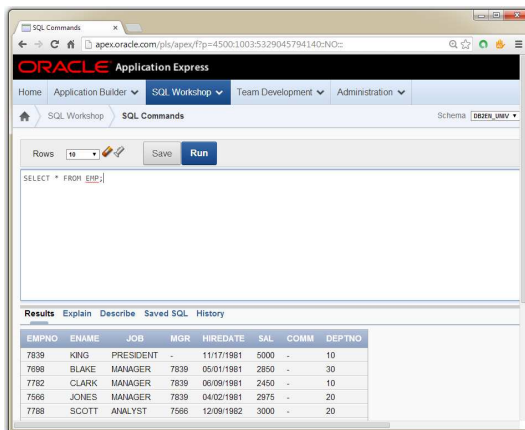
- The Object Browser – used to browse the DB objects, and to view and modify interactively their properties.
- The SQL Commands – used to create, edit, view, run, and delete database objects through SQL operations.
- The SQL Scripts - a SQL script is a set of SQL commands saved as a file in SQL Scripts. This component will manage such scripts.
- The Utilities package – is used to build SQL queries, load and unload data from an Oracle database, generate DDL, view object reports, manage User Interface Defaults, restore dropped database objects, compare schemas, monitor the database, and view database details.
- The RESTful Services – enable declarative specification of RESTful services that can be later used to access the database.



**ACTIVITY 1:** Using the SQL Workshop / Object Browser component try to visualize both structure and data of all tables created in the previous chapter.



 **ACTIVITY 2:** Using the SQL Workshop / SQL Commands component try to visualize the content of tables based on **SELECT** SQL command.



Note: use the Rows selector to modify the number of records displayed in the Results window.

### 1.3. Inserting, updating and deleting data

The **INSERT** SQL statement is used to insert new records in a table. Several syntax are supported:

**INSERT INTO** table **VALUES** (value1, value2, ...);

---

Values have to corresponds 1:1 to table columns from the schema definition. Therefore, the number of values in the list must be the same as relation grade.

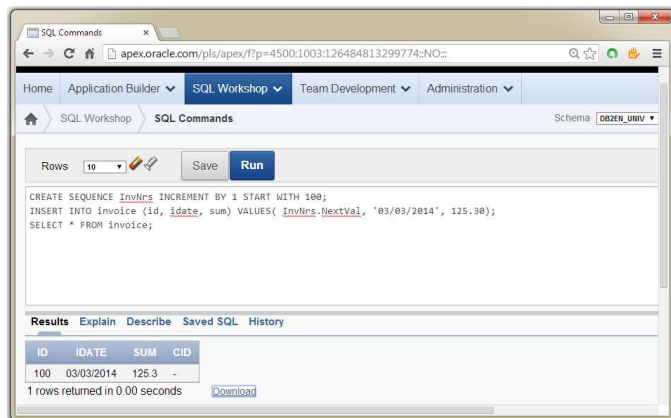
**INSERT INTO** table (column1, column2, ...)  
**VALUES** (value1, value2, ...);


Using this syntax only a specific sequence of columns has to be filled. However, it has to include all attributes that have a NOT NULL associated constraint (including PRIMARY KEY).

**INSERT INTO** table (column1, column2, ... )  
**SELECT** expression1, expression2, ... **FROM** source\_table;

This syntax is used to copy data from a source table to another table. The column1 will be assigned the value of expression1, the column2 will be assigned the value of expression2, and so on. The number and types of expressions should correspond to the column list.

**CREATE SEQUENCE** could be used in conjunction with **INSERT** to generate a sequence of unique numeric values into a primary key.



 **ACTIVITY 3:** In the SQL Workshop / SQL Commands component add 5 students to the University database from the previous chapter. Try to use both syntaxes for **INSERT**,

---

with and without column list. What happened if the primary key is missing in the column list?

The **UPDATE** statement is used to update records in a table.

**UPDATE** table

**SET** column1=value1, column2=value2, ...

**WHERE** logical\_condition;

The logical condition is a logical expression composed by table fields, constants, comparison operators (=, !=, <, <=, >, >=) and logical connectors (AND, OR). The **WHERE** clause specifies which record or records will be updated. If **WHERE** clause is missing, **all records will be updated!** A condition involving PRIMARY KEY has to be used to modify a single record.

E.g. **UPDATE** invoice

**SET** sum=sum\*1.24

**WHERE** id=100;



**ACTIVITY 4:** In the SQL Workshop / SQL Commands component, update students from the University database in order to go into the next year of study. Try to write then a **UPDATE** statement that undo all changes of the year values. What happened if you change only the students from the first year?

The **DELETE** statement is used to delete records in a table.

**DELETE FROM** table

**WHERE** logical\_condition;

The **WHERE** clause specifies which record or records will be deleted. If you omit the **WHERE** clause, all records will be deleted!



**ACTIVITY 5:** Delete all students from the final year. Note: there is no **UNDO** operation (in absence of transactions).