Dated:

SQL: (Structured Query Language).

A Non procedural language which is used to communicate between data analyst and databases. functions of DBMS is written in SQL. SQL only works for Relational Database system RDBMS. Note that SQL isn't case sensitive language. some manipulated types of SQL are:-

TSQL (Transaction Structured Query Language)

PSQL (Programming Language Structured Query Language)

SQL Command Types

DDL	DCL	DQL	DML
Data Definition Language	Data Control Language	Data Query Language	Data Manipulation Language
* Create	* Grant	* Select	* Insert
* Alter	* Revoke		* Update
* Drop			* Delete
* Rename			* Merge

COMMANDS	PARAMETERS
SELECT	FIELDS-LIST / * / aggregate function
FROM	Table name
WHERE	condition
GROUPBY	FIELD NAME
ORDER BY	FIELD NAME [DESCRIPTION] (-SENDR NOTES).

Note that:

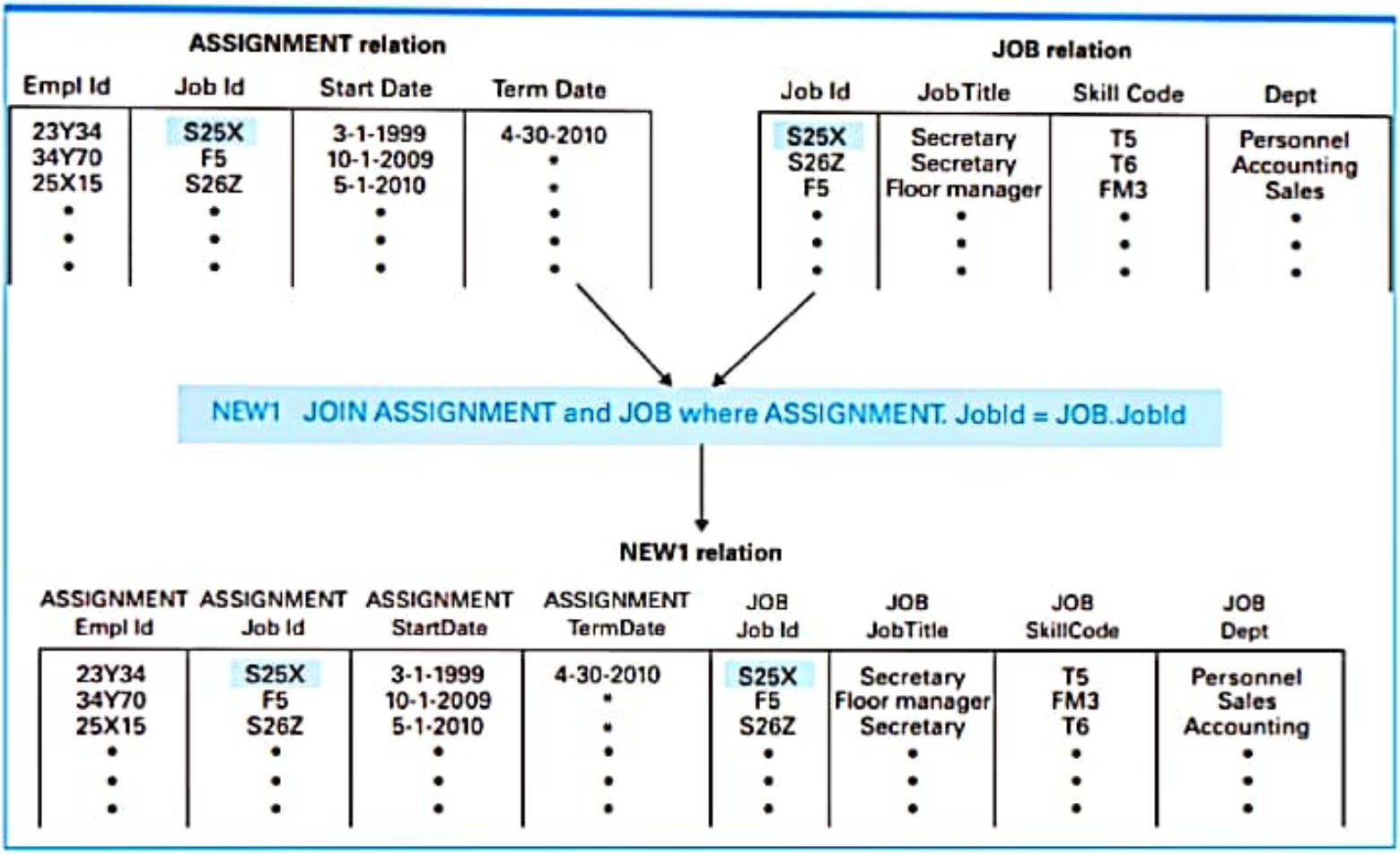
MY SQL is the enviournment which is connected with server by data base engine.

Scanned with CamScanner

SQL

Now that we have introduced the basic relational operations, let us reconsider the overall structure of a database system. Remember that a database is actually stored in a mass storage system. To relieve the application programmer from the details of such systems, a database management system is provided that allows the application software to be written in terms of a database model, such as the relational model. The DBMS accepts commands in terms of the model and converts them into actions relative to the actual storage structure. This conversion is handled by a collection of routines within the DBMS that are used by the application software as abstract tools. Thus a DBMS based on the relational model would include routines to perform the SELECT, PROJECT, and JOIN operations, which could then be called from the application software. In this manner the

Figure 9.12 An application of the JOIN operation



Scanned with CamScanner

application software can be written as though the database were actually stored in the simple tabular form of the relational model.

Today's relational database management systems do not necessarily provide routines to perform the SELECT, PROJECT, and JOIN operations in their raw form. Instead, they provide routines that might be combinations of these basic steps. An example is the language SQL (Structured Query Language), which forms the backbone of most relational database query systems. For example, SQL is the underlying language in the relational database system MySQL (pronounced "My-S-Q-L") used by many database servers in the Internet.

One reason for SQL's popularity is that the American National Standards Institute has standardized it. Another reason is that it was originally developed and marketed by IBM and has thus benefited from a high level of exposure. In this section we explain how relational database queries are expressed in SQL.

Although we are about to find that a query stated in SQL is expressed in an imperative-sounding form, the reality is that it is essentially a declarative statement. You should read an SQL statement as a description of the information desired rather than a sequence of activities to be performed. The significance of this is that SQL relieves application programmers from the burden of developing algorithms for manipulating relations—they need merely to describe the information desired.

For our first example of an SQL statement, let us reconsider our last query in which we developed a three-step process for obtaining all employee identification numbers along with their corresponding departments. In SQL this entire query could be represented by the single statement

```
select EmplId, Dept
from ASSIGNMENT, JOB
where ASSIGNMENT.JobId = JOB.JobId
  and ASSIGNMENT.TermDate = '*'
```

As indicated by this example, each SQL query statement can contain three clauses: a `select` clause, a `from` clause, and a `where` clause. Roughly speaking, such a statement is a request for the result of forming the JOIN of all the relations listed in the `from` clause, SELECTing those tuples that satisfy the conditions in the `where` clause, and then PROJECTing those tuples listed in the `select` clause. (Note that the terminology is somewhat reversed since the `select` clause in an SQL statement identifies the attributes used in the PROJECT operation.) Let us consider some simple examples.

The statement

```
select Name, Address
from EMPLOYEE
```

produces a listing of all employee names and addresses contained in the relation EMPLOYEE. Note that this is merely a PROJECT operation.

The statement

```
select EmplId, Name, Address, SSNum
from EMPLOYEE
where Name = 'Cheryl H. Clark'
```

produces all the information from the tuple associated with Cheryl H. Clark in the EMPLOYEE relation. This is essentially a SELECT operation.

Scanned with CamScanner

The statement

```
select Name, Address
from EMPLOYEE
where Name = 'Cheryl H. Clark'
```

produces the name and address of Cheryl H. Clark as contained in the EMPLOYEE relation. This is a combination of SELECT and PROJECT operations.

The statement

```
select EMPLOYEE.Name, ASSIGNMENT.StartDate
from EMPLOYEE, ASSIGNMENT
where EMPLOYEE.EmplId = ASSIGNMENT.EmplId
```

produces a listing of all employee names and their dates of initial employment. Note that this is the result of JOINing the relations EMPLOYEE and ASSIGNMENT and then SELECTing and PROJECTing the appropriate tuples and attributes as identified in the `where` and `select` clauses.

We close by noting that SQL encompasses statements for defining the structure of relations, creating relations, and modifying the contents of relations as well as performing queries. For example, the following are examples of the `insert into`, `delete from`, and `update` statements.

The statement

```
insert into EMPLOYEE
values ('42Z12', 'Sue A. Burt', '33 Fair St.',
        '444661111')
```

adds a tuple to the EMPLOYEE relation containing the values given;

```
delete from EMPLOYEE
where Name = 'G. Jerry Smith'
```

removes the tuple relating to G. Jerry Smith from the EMPLOYEE relation; and

```
update EMPLOYEE
set Address = '1812 Napoleon Ave.'
where Name = 'Joe E. Baker'
```

changes the address in the tuple associated with Joe E. Baker in the EMPLOYEE relation.

Questions & Exercises

1. Answer the following questions based on the partial information given in the EMPLOYEE, JOB, and ASSIGNMENT relations in Figure 9.5:
   a. Who is the secretary in the accounting department with experience in the personnel department?
   b. Who is the floor manager in the sales department?
   c. What job does G. Jerry Smith currently hold?
2. Based on the EMPLOYEE, JOB, and ASSIGNMENT relations presented in Figure 9.5, write a sequence of relational operations to obtain a list of all job titles within the personnel department.

3. Based on the EMPLOYEE, JOB, and ASSIGNMENT relations presented in Figure 9.5, write a sequence of relational operations to obtain a list of employee names along with the employees' departments.
4. Convert your answers to Questions 2 and 3 into SQL.
5. How does the relational model provide for data independence?
6. How are the different relations in a relational database tied together?

Scanned with CamScanner