

# Introduction to Views

About this chapter

It is often impractical to repeatedly type complicated queries. SQL provides **views**, that allow you to give names to frequently executed SELECT commands. A view is a computed table which is useful for security purposes, and for tailoring the appearance of database information to make data access straightforward.

This chapter introduces you to views.

Contents

<b>Topic</b>	<b>Page</b>
Defining a view	272
Using views for security	274

## Defining a view

Suppose that you frequently need to list a summary of employees and the departments they belong to. The following query produces the information you need.

❖ **To list employees and the departments to which they belong:**

- ◆ Type the following:

```
SELECT emp_fname, emp_lname, dept_name
FROM employee JOIN department
ON department.dept_id = employee.dept_id
```

emp_fname	emp_lname	department
Fran	Whitney	R&D
Matthew	Cobb	R&D
Robert	Breault	R&D
Natasha	Shishov	R&D
Kurt	Driscoll	R&D

You can create a view that produces the results of this command as follows:

```
CREATE VIEW emp_dept AS
SELECT emp_fname, emp_lname, dept_name
FROM employee JOIN department
ON department.dept_id = employee.dept_id
```

This command creates a view called **emp\_dept** that looks in many respects just like any other table in the database.

You can list everything in this view just as you do from a table:

❖ **To list employees and the departments to which they belong:**

- ◆ Type the following:

```
SELECT *
FROM emp_dept
```

emp_fname	emp_lname	department
Fran	Whitney	R&D
Matthew	Cobb	R&D
Robert	Breault	R&D
Natasha	Shishov	R&D
Kurt	Driscoll	R&D

It is important to remember that the information in a view is not stored separately in the database. Each time you refer to the view, SQL executes the associated SELECT statement to find the appropriate data.

On one hand, this is good; it means that if someone modifies the **employee** table or the **department** table, the information in the **emp\_dept** view will be automatically up to date. On the other hand, if the SELECT command is complicated it may take a long time for SQL to find the correct information every time you use the view.

#### Providing names for the view columns

You can provide names for the view columns explicitly. First you must get rid of the original view definition as follows:

```
DROP VIEW emp_dept
```

You can redefine the view with the new column name as follows:

```
CREATE VIEW emp_dept(FirstName, LastName, Department) AS
SELECT emp_fname, emp_lname, dept_name
FROM employee JOIN department
ON department.dept_id = employee.dept_id
```

You have changed the names of the columns in the view by specifying new column names in parentheses after the view name. Do SELECT \* again to see the new column names.

#### More about views

Views can be thought of as computed tables. Any SELECT command can be used in a view definition except commands containing ORDER BY. Views can use GROUP BY clauses, subqueries, and joins. Disallowing ORDER BY is consistent with the fact that rows of a table in a relational database are not stored in any particular order. When you use the view, you can specify an ORDER BY.

You can also use views in more complicated queries. Here is an example using a join:

```
SELECT LastName, dept_head_id
FROM emp_dept, department
WHERE emp_dept.Department = department.dept_name
```

## Using views for security

### Example

Views can be used to restrict access to information in the database. For example, suppose you wanted to create a user ID for the sales department head, Moira Kelly, and restrict her user ID so that it can only examine information about employees in the sales department.

### Creating the new user ID

First you need to create the new user ID for Moira Kelly using the GRANT statement. From Interactive SQL, connected to the sample database as **dba**, enter the following:

```
GRANT CONNECT TO M_Kelly  
IDENTIFIED BY SalesHead
```

### Granting permissions

Next you need to grant user **M\_Kelly** the right to look at employees of the sales department.

```
CREATE VIEW SalesEmployee AS  
SELECT emp_id, emp_lname, emp_fname  
FROM "dba".employee  
WHERE dept_id = 200
```

The table should be identified as **"dba".employee** for the **M\_Kelly** user ID to be able to use the view.

Now you must give **M\_Kelly** permission to look at the new view by entering:

```
GRANT SELECT ON SalesEmployee TO M_Kelly
```

### Looking at the view

Connect to the database as **M\_Kelly** and now try looking at the view:

```
CONNECT USER M_Kelly IDENTIFIED BY SalesHead ;  
SELECT * FROM "dba".SalesEmployee
```

emp_id	emp_lname	emp_fname
129	Chin	Philip
195	Dill	Marc
299	Overbey	Rollin
467	Klobucher	James
641	Powell	Thomas

However, you do not have permission to look directly at the **employee** and **department** tables. If you execute the following commands, you will get permission errors.

```
SELECT * FROM "dba".employee ;  
SELECT * FROM "dba".department
```

**Notes**

Make sure you connect back to the sample database before proceeding to the next chapter.

