CREATE TABLE department (  
Did INTEGER PRIMARY KEY,  
Dname CHAR (30),  
Dacctno CHAR (6) )

CREATE TABLE employee (  
Ename CHAR (30),  
Did INTEGER,  
SalaryCode INTEGER,  
Eid INTEGER PRIMARY KEY,  
Ephone CHAR (10),  
FOREIGN KEY (Did) REFERENCES department (Did) )

The basic command for retrieving information is the SELECT statement. Consider this example:

SELECT Ename, Eid, Ephone  
FROM Employee  
WHERE Did = 15

This query returns the Ename, Eid, and Ephone fields from the Employee table for all employees assigned to department 15.

The view in Figure 5.3b is created using the following SQL statement:

CREATE VIEW newtable (Dname, Ename, Eid, Ephone)  
AS SELECT D.Dname E.Ename, E.Eid, E.Ephone  
FROM Department D Employee E  
WHERE E.Did = D.Did
GRANT { privileges | role }
[ON table]
TO { user | role | PUBLIC }
[WITH GRANT OPTION]

Different implementations of SQL provide different ranges of access rights. The following is a typical list:

- **Select**: Grantee may read entire database; individual tables; or specific columns in a table.
- **Insert**: Grantee may insert rows in a table; or insert rows with values for specific columns in a table.
- **Update**: Semantics is similar to INSERT.
- **Delete**: Grantee may delete rows from a table.
- **References**: Grantee is allowed to define foreign keys in another table that refer to the specified columns.

The REVOKE command has the following syntax:

REVOKE { privileges | role }
[ON table]
FROM { user | role | PUBLIC }
CREATE view V1 AS
SELECT Position, Salary
FROM Employee
WHERE Department = "strip"

CREATE view V2 AS
SELECT Name, Department
FROM Employee
WHERE Department = "strip"

Users of these views are not authorized to access the relationship between Name and Salary.

A user who has access to either or both views cannot infer the relationship by functional dependencies.

But a user who knows the structure of the Employee table and who knows that the view tables maintain the same row order as the Employee table is then able to merge the two views to construct the table shown in Figure 5.6c.
Notes for Slide #17

**Stopping inference by database design is non-trivial. Consider the following example.**

Individually, the name, address, and salary information is available to a subordinate role, such as Clerk, but the association of names and salaries is restricted to a superior role, such as Administrator.

One approach is to construct three tables, which include the following information:

- Employees (Emp#, Name, Address)
- Salaries (S#, Salary)
- Emp-Salary (Emp#, S#)

where each line consists of the table name followed by a list of column names for that table. In this case, each employee is assigned a unique employee number (Emp#) and a unique salary number (S#).

The Employees table and the Salaries table are accessible to the Clerk role, but the Emp-Salary table is only available to the Administrator role. In this structure, the sensitive relationship between employees and salaries is protected from users assigned the Clerk role.

Now suppose that we want to add a new attribute, employee start date, which is not sensitive. This could be added to the Salaries table as follows:

- Employees (Emp#, Name, Address)
- Salaries (S#, Salary, Start-Date)
- Emp-Salary (Emp#, S#)

However, an employee’s start date is an easily observable or discoverable attribute of an employee. Thus a user in the Clerk role should be able to infer (or partially infer) the employee’s name. This would compromise the relationship between employee and salary.

A straightforward way to remove the inference channel is to add the start-date column to the Employees table rather than to the Salaries table.

**Stopping inference at query time is also non-trivial. Consider the following example.**

I know that to work on this top secret project you need access to A, B, and C. I query for all users with access to A, then all users with access to B, then all users with access to C. The intersection is the people that may be on the top secret project.

Need to keep track of past queries and decide whether the new query will allow inference...