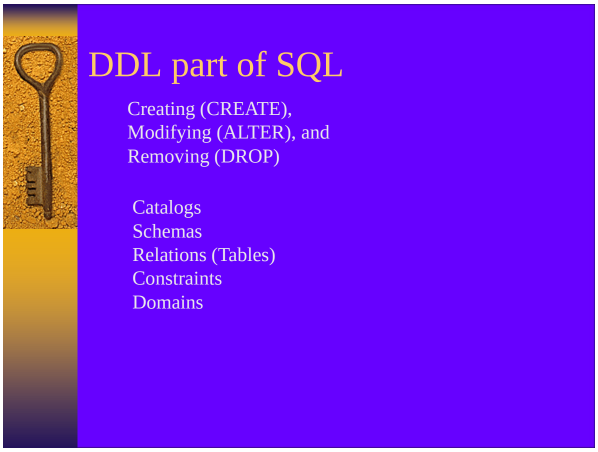
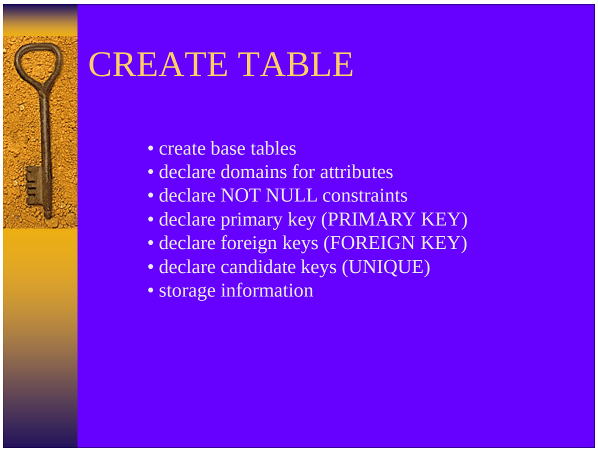


DDL part of SQL

Creating (CREATE),
Modifying (ALTER), and
Removing (DROP)

Catalogs
Schemas
Relations (Tables)
Constraints
Domains

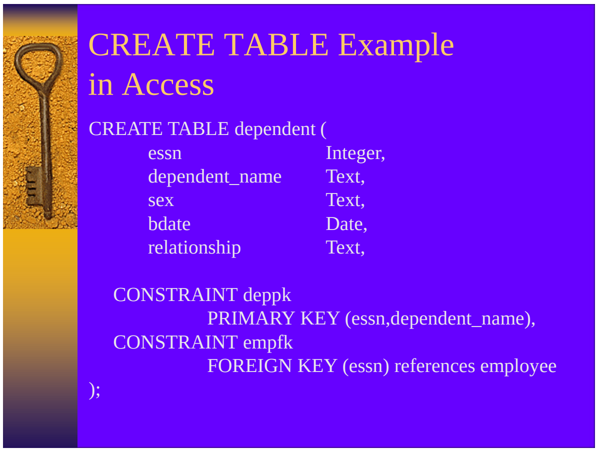


CREATE TABLE

- create base tables
- declare domains for attributes
- declare NOT NULL constraints
- declare primary key (PRIMARY KEY)
- declare foreign keys (FOREIGN KEY)
- declare candidate keys (UNIQUE)
- storage information

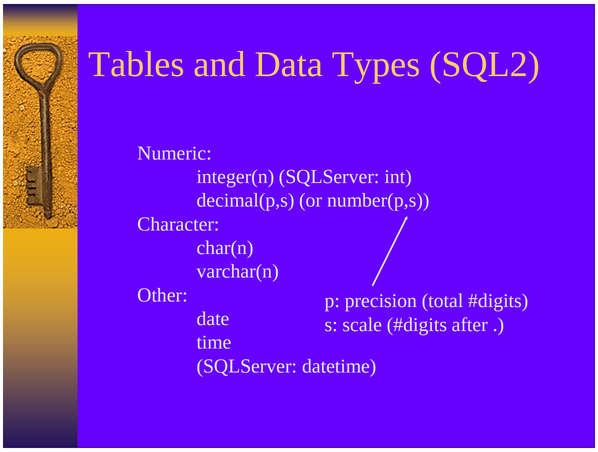
CREATE TABLE Example in Access

```
CREATE TABLE department (
        dname Text UNIQUE NOT NULL,
        dnumber Integer PRIMARY KEY,
        mgrssn Integer REFERENCES employee,
        mgrstartdate Date
);
```



CREATE TABLE Example in Access

```
CREATE TABLE dependent (
        essn                    Integer,
        dependent_name          Text,
        sex                     Text,
        bdate                   Date,
        relationship            Text,

   CONSTRAINT deppk
               PRIMARY KEY (essn,dependent_name),
   CONSTRAINT empfk
               FOREIGN KEY (essn) references employee
);
```



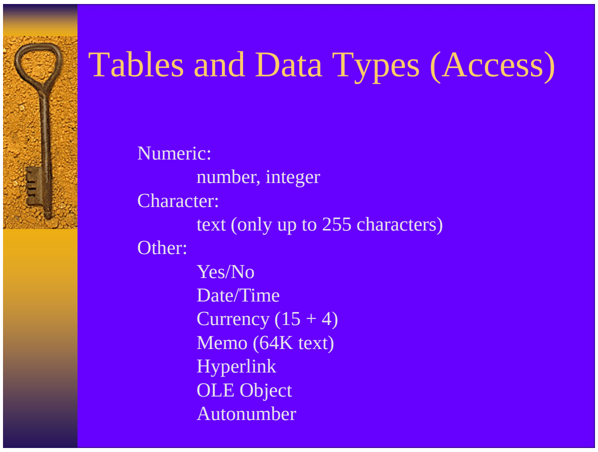
Tables and Data Types (SQL2)

Numeric:
- integer(n) (SQLServer: int)
- decimal(p,s) (or number(p,s))

Character:
- char(n)
- varchar(n)

Other:
- date
- time
- (SQLServer: datetime)

p: precision (total #digits)
s: scale (#digits after .)

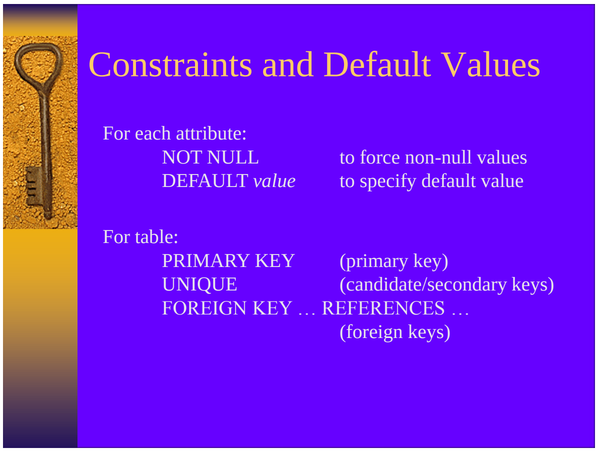


Tables and Data Types (Access)

Numeric:
- number, integer

Character:
- text (only up to 255 characters)

Other:
- Yes/No
- Date/Time
- Currency (15 + 4)
- Memo (64K text)
- Hyperlink
- OLE Object
- Autonumber



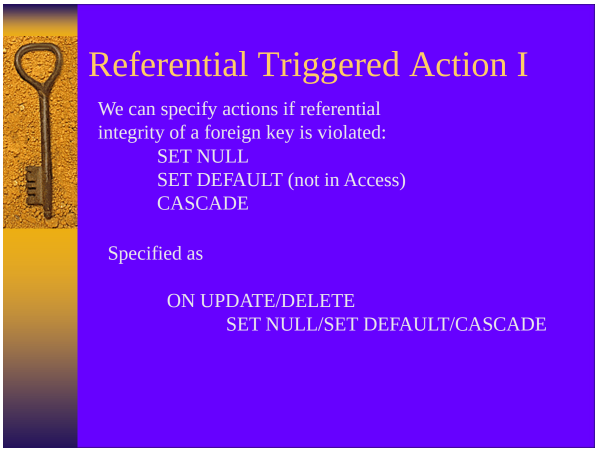
Constraints and Default Values

For each attribute:

NOT NULL	to force non-null values
DEFAULT *value*	to specify default value

For table:

PRIMARY KEY	(primary key)
UNIQUE	(candidate/secondary keys)
FOREIGN KEY … REFERENCES …	(foreign keys)



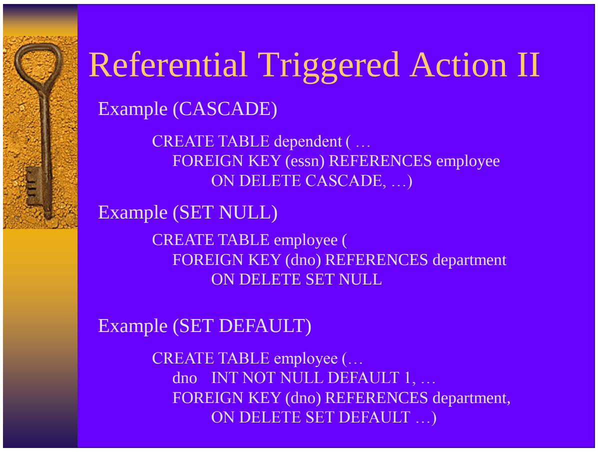
Referential Triggered Action I

We can specify actions if referential integrity of a foreign key is violated:

SET NULL
SET DEFAULT (not in Access)
CASCADE

Specified as

ON UPDATE/DELETE
SET NULL/SET DEFAULT/CASCADE



Referential Triggered Action II

Example (CASCADE)

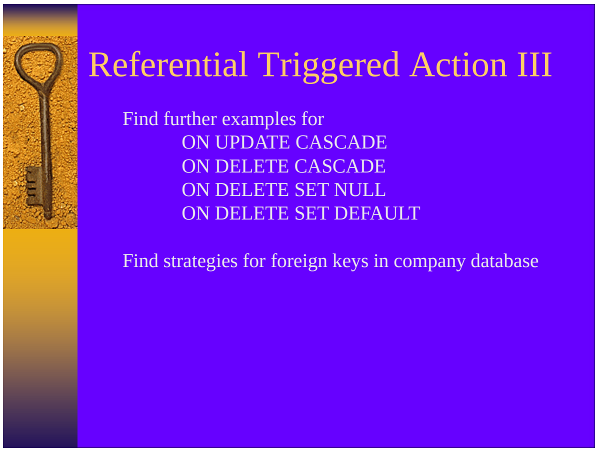
```
CREATE TABLE dependent ( …
   FOREIGN KEY (essn) REFERENCES employee
         ON DELETE CASCADE, …)
```

Example (SET NULL)

```
CREATE TABLE employee (
   FOREIGN KEY (dno) REFERENCES department
         ON DELETE SET NULL
```

Example (SET DEFAULT)

```
CREATE TABLE employee (…
   dno   INT NOT NULL DEFAULT 1, …
   FOREIGN KEY (dno) REFERENCES department,
         ON DELETE SET DEFAULT …)
```



Referential Triggered Action III

Find further examples for

```
ON UPDATE CASCADE
ON DELETE CASCADE
ON DELETE SET NULL
ON DELETE SET DEFAULT
```

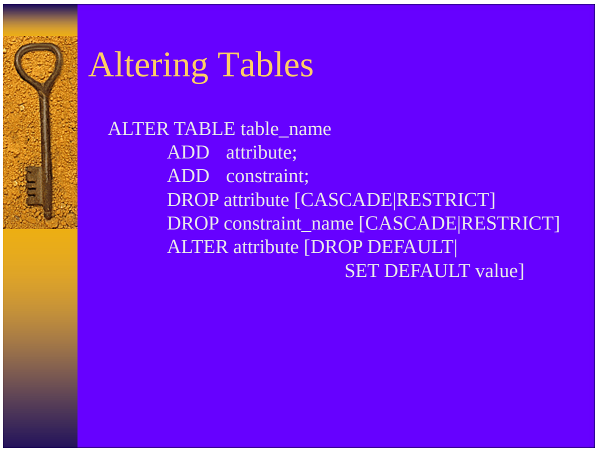
Find strategies for foreign keys in company database

Dropping Tables

Drop behaviors: cascade and restrict

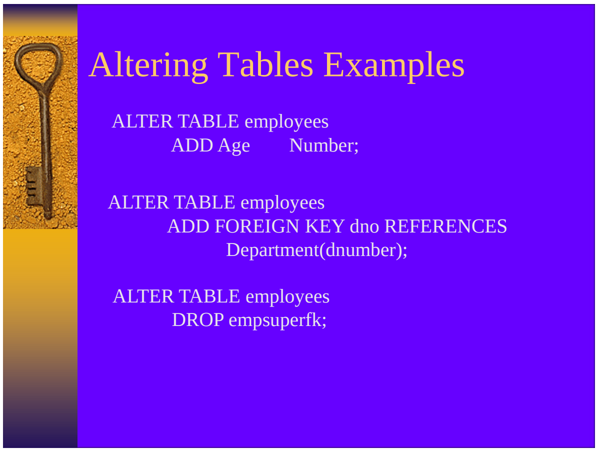
```
DROP TABLE Dependent CASCADE;

DROP TABLE Dependent RESTRICT;
```

only drops if no element of table is referenced



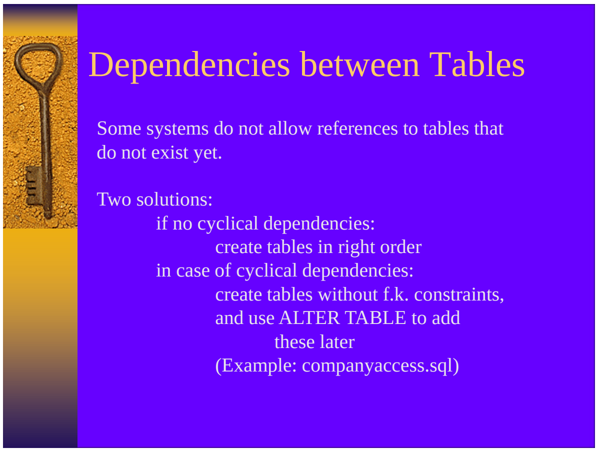
Altering Tables

```
ALTER TABLE table_name
        ADD   attribute;
        ADD   constraint;
        DROP attribute [CASCADE|RESTRICT]
        DROP constraint_name [CASCADE|RESTRICT]
        ALTER attribute [DROP DEFAULT|
                                SET DEFAULT value]
```



Altering Tables Examples

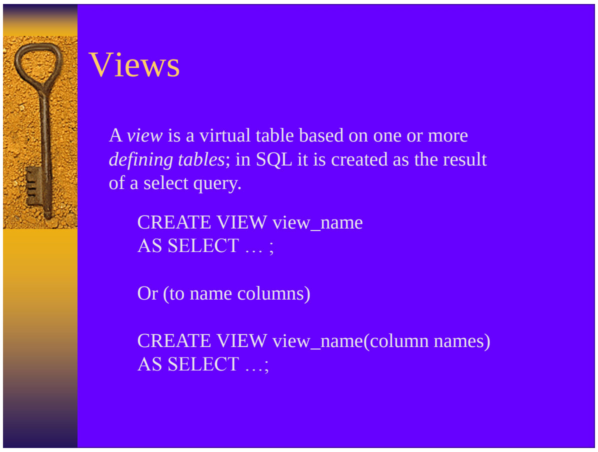
```
ALTER TABLE employees
        ADD Age       Number;

ALTER TABLE employees
        ADD FOREIGN KEY dno REFERENCES
                Department(dnumber);

ALTER TABLE employees
        DROP empsuperfk;
```



Dependencies between Tables

Some systems do not allow references to tables that do not exist yet.

Two solutions:
        if no cyclical dependencies:
                create tables in right order
        in case of cyclical dependencies:
                create tables without f.k. constraints,
                and use ALTER TABLE to add
                        these later
                (Example: companyaccess.sql)



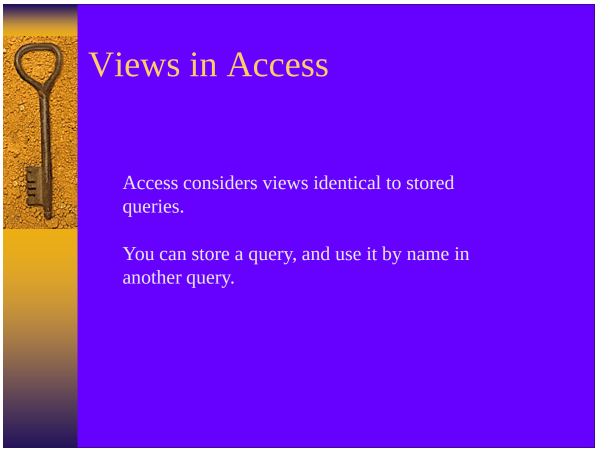
Views

A *view* is a virtual table based on one or more *defining tables*; in SQL it is created as the result of a select query.

```
CREATE VIEW view_name
AS SELECT … ;
```

Or (to name columns)

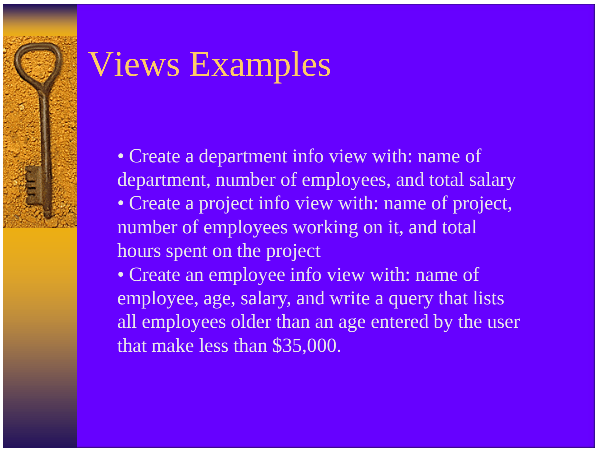
```
CREATE VIEW view_name(column names)
AS SELECT …;
```



Views in Access

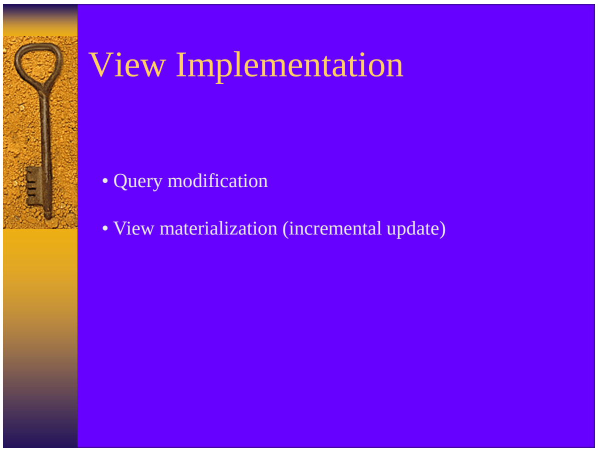
Access considers views identical to stored queries.

You can store a query, and use it by name in another query.



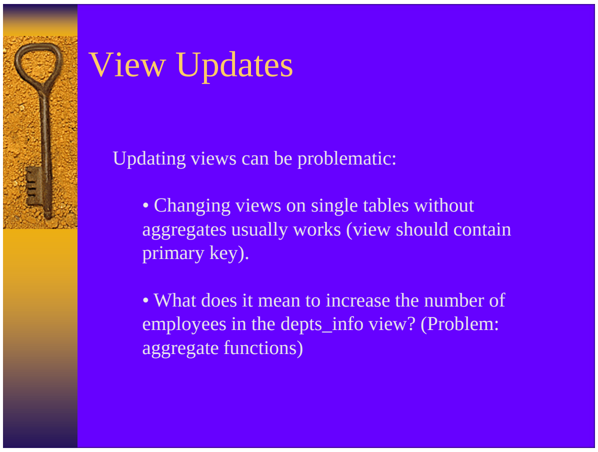
Views Examples

- Create a department info view with: name of department, number of employees, and total salary
- Create a project info view with: name of project, number of employees working on it, and total hours spent on the project
- Create an employee info view with: name of employee, age, salary, and write a query that lists all employees older than an age entered by the user that make less than $35,000.



View Implementation

- Query modification
- View materialization (incremental update)



View Updates

Updating views can be problematic:

- Changing views on single tables without aggregates usually works (view should contain primary key).
- What does it mean to increase the number of employees in the depts_info view? (Problem: aggregate functions)