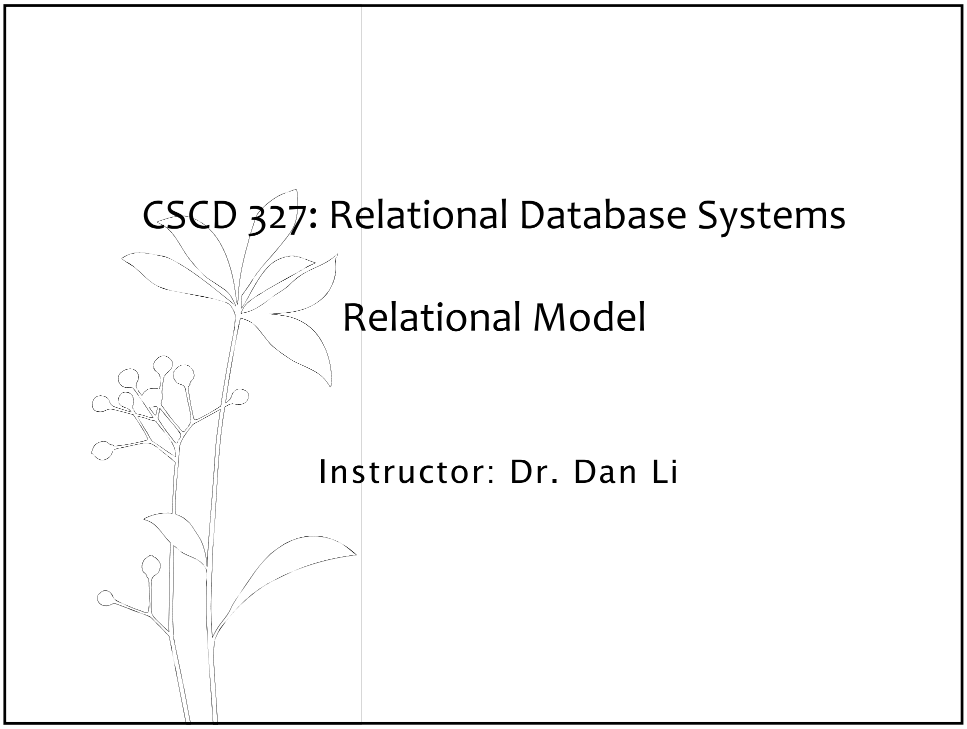
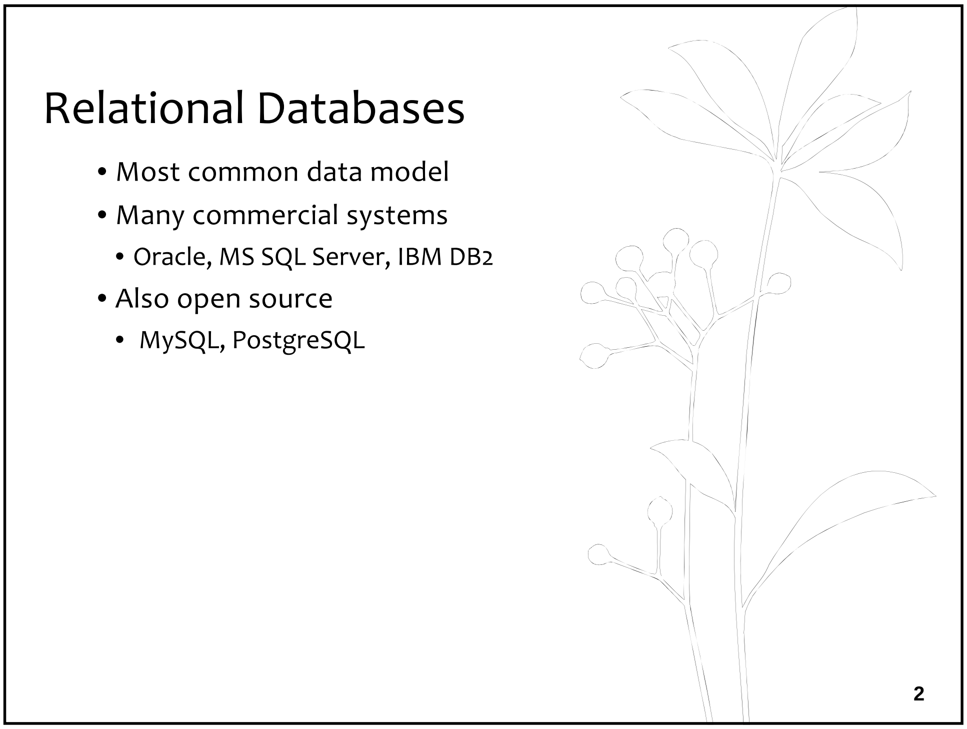


CSCD 327: Relational Database Systems

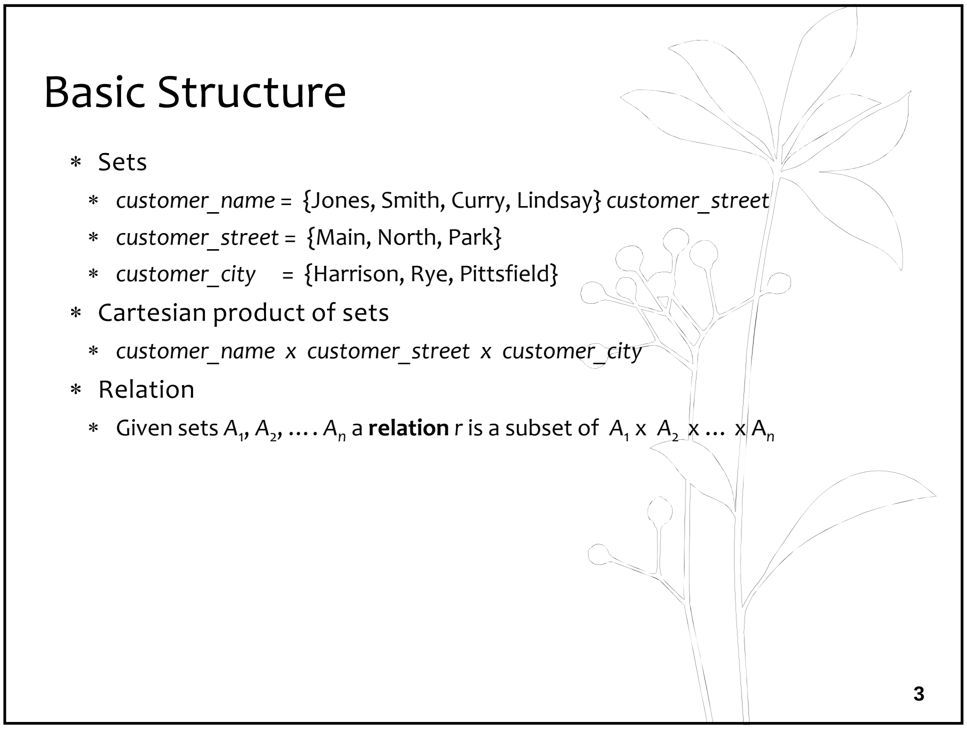
Relational Model

Instructor: Dr. Dan Li



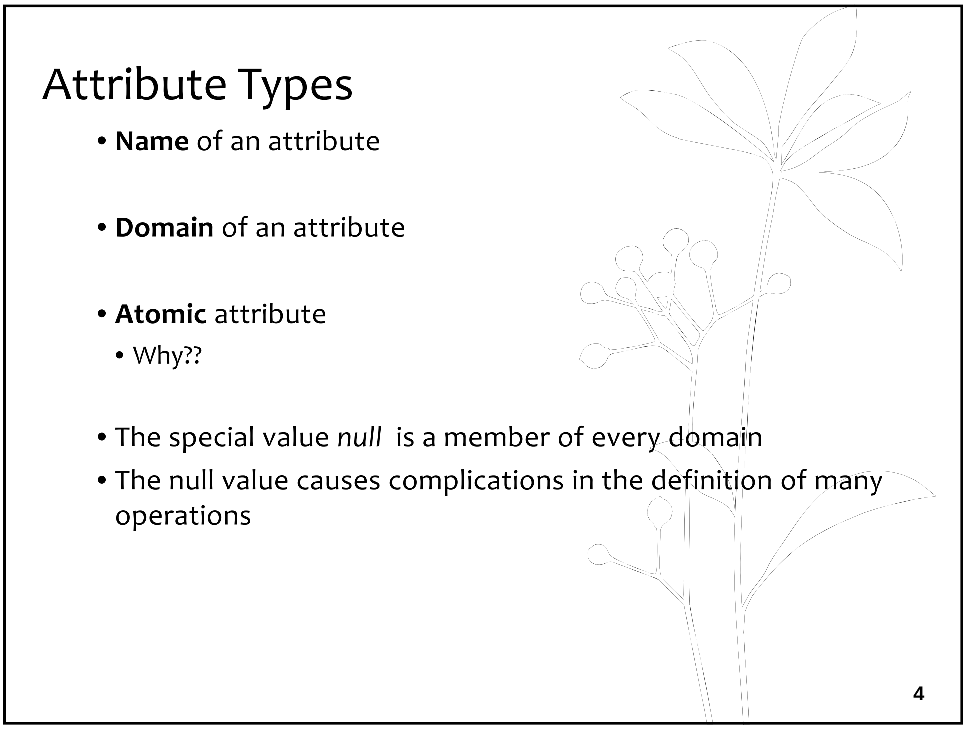
Relational Databases

- Most common data model
- Many commercial systems
  - Oracle, MS SQL Server, IBM DB2
- Also open source
  - MySQL, PostgreSQL



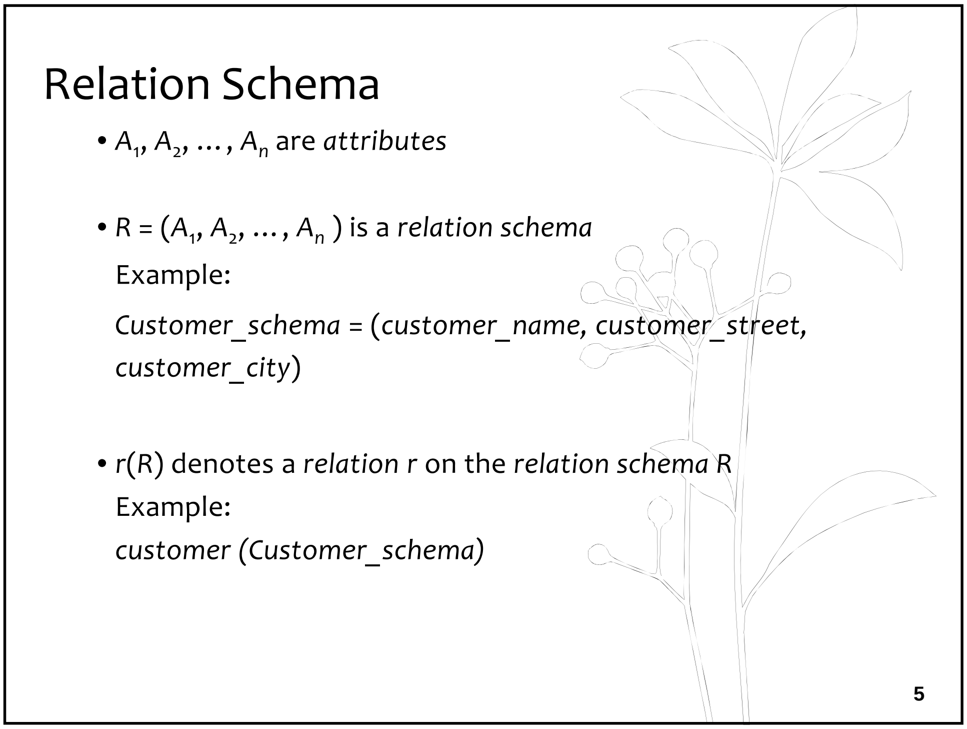
Basic Structure

- Sets
  - *customer_name* = {Jones, Smith, Curry, Lindsay} *customer_street*
  - *customer_street* = {Main, North, Park}
  - *customer_city* = {Harrison, Rye, Pittsfield}
- Cartesian product of sets
  - *customer_name* x *customer_street* x *customer_city*
- Relation
  - Given sets A_1, A_2, ... A_n a **relation** *r* is a subset of A_1 x A_2 x ... x A_n



Attribute Types

- **Name** of an attribute
- **Domain** of an attribute
- **Atomic** attribute
  - Why??
- The special value *null* is a member of every domain
- The null value causes complications in the definition of many operations



Relation Schema

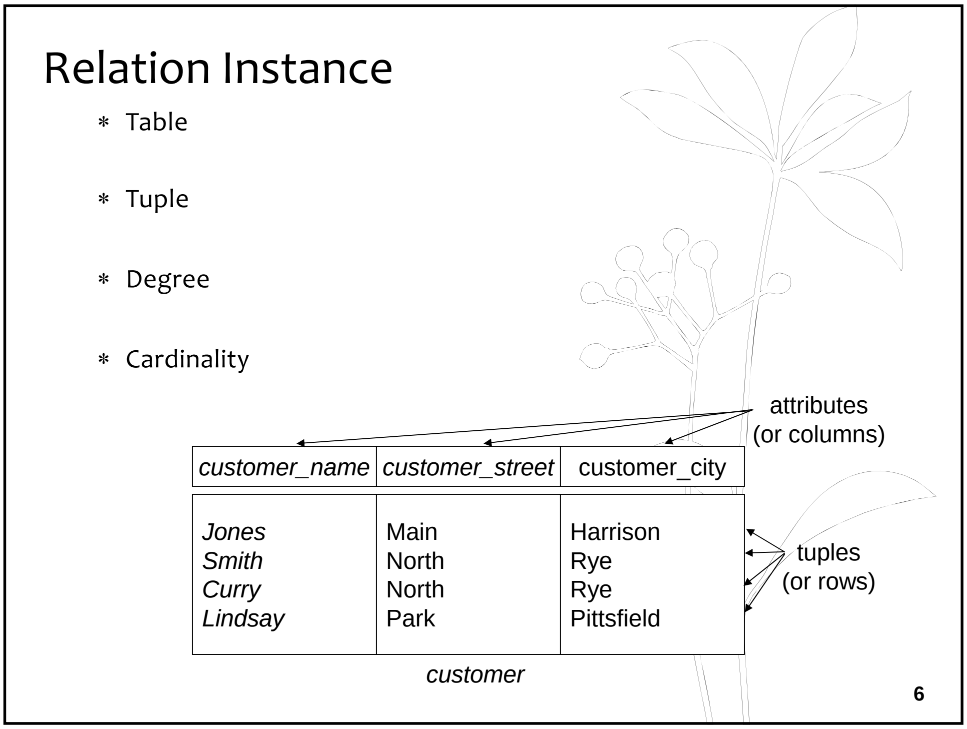
- $A_1, A_2, \ldots, A_n$ are *attributes*
- $R = (A_1, A_2, \ldots, A_n)$ is a *relation schema*

  Example:

  Customer_schema = (*customer_name, customer_street, customer_city*)
- r(*R*) denotes a *relation r* on the *relation schema* R

  Example:

  customer (Customer_schema)



Relation Instance

∗ Table

∗ Tuple

∗ Degree

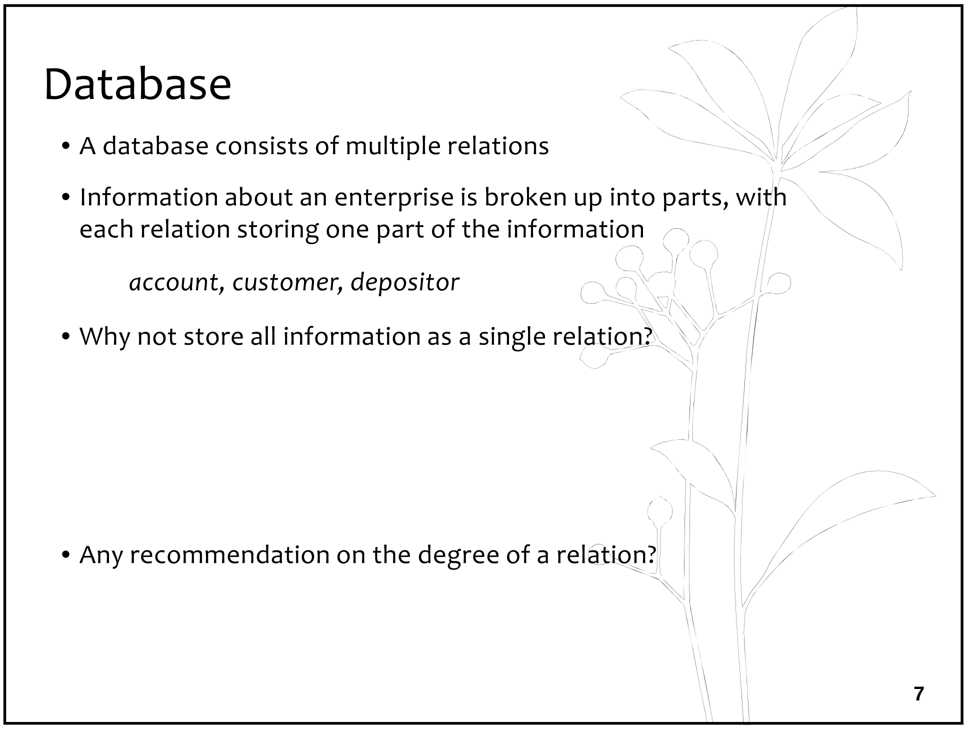
∗ Cardinality

attributes
(or columns)

customer_name	*customer_street*	customer_city
Jones	Main	Harrison
Smith	North	Rye
Curry	North	Rye
Lindsay	Park	Pittsfield

tuples
(or rows)

customer

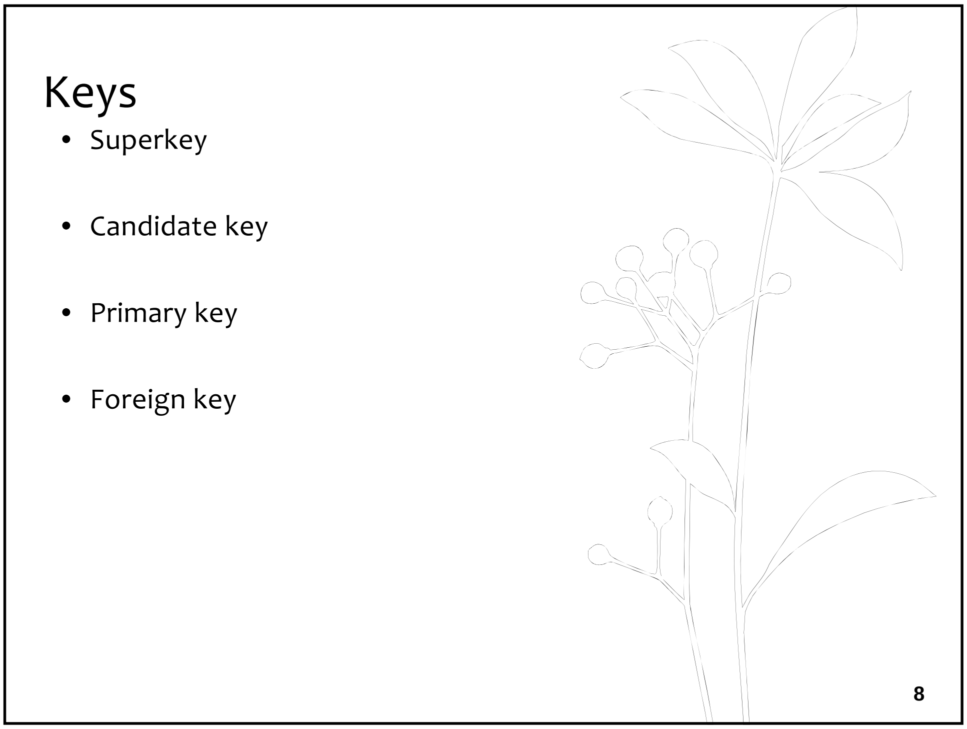


Database

- A database consists of multiple relations
- Information about an enterprise is broken up into parts, with each relation storing one part of the information

  account, customer, depositor
- Why not store all information as a single relation?
- Any recommendation on the degree of a relation?

7



Keys

- Superkey
- Candidate key
- Primary key
- Foreign key

8

SQL Data Definition Language (DDL)

- SQL is primarily a query language, for getting information from a database.
- But SQL also includes a *data-definition* component for describing database schemas.
- **CREATE TABLE**
- **ALTER TABLE**
- **DROP TABLE**

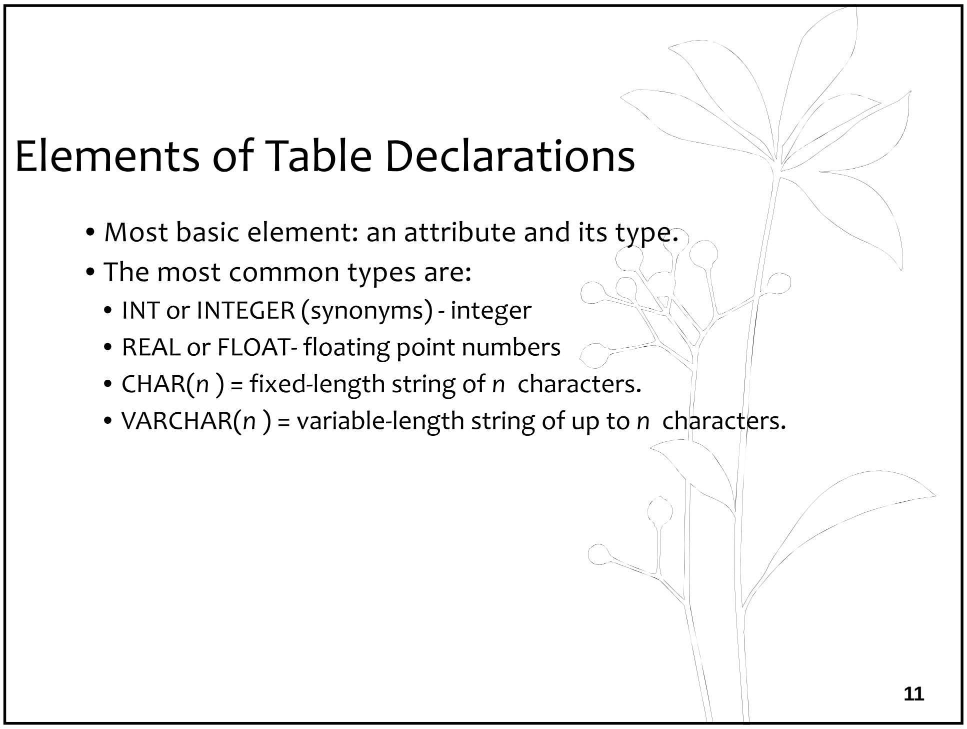
Creating (Declaring) a Relation

- Simplest form is:

```
CREATE TABLE <name> (
        <list of elements>
);
```

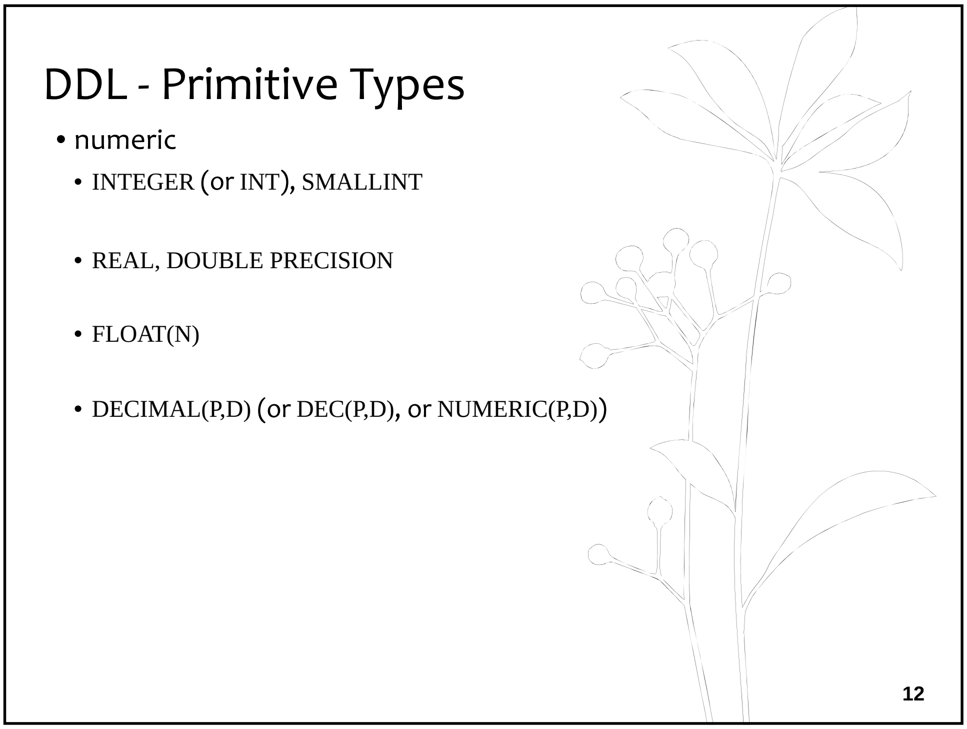
- To delete a relation:

```
DROP TABLE <name>;
```



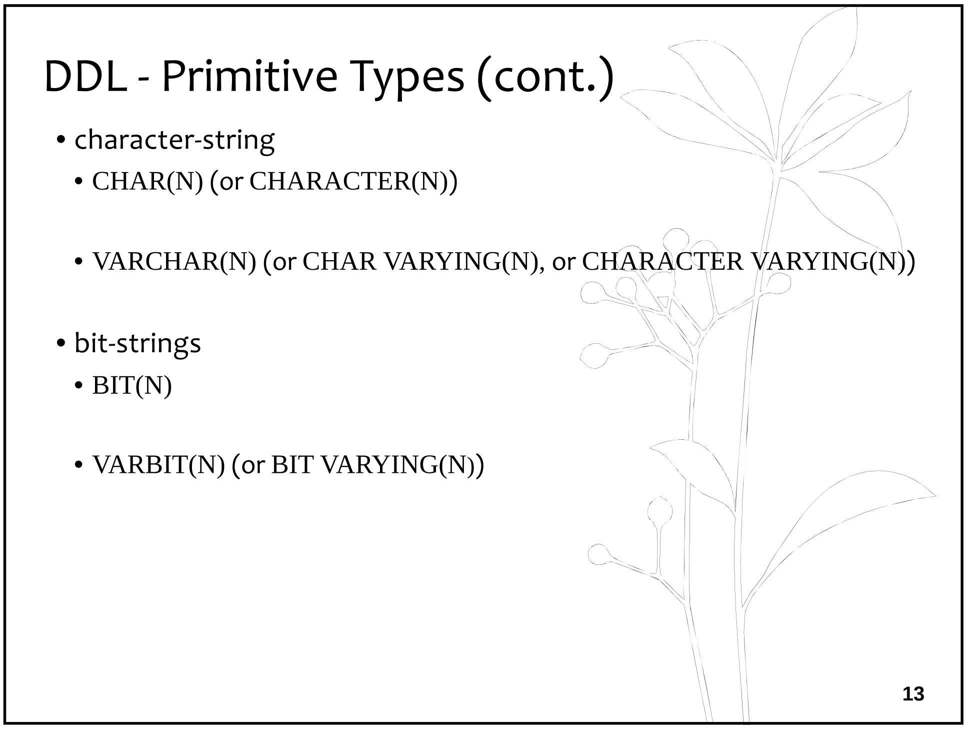
Elements of Table Declarations

- Most basic element: an attribute and its type.
- The most common types are:
  - INT or INTEGER (synonyms) - integer
  - REAL or FLOAT- floating point numbers
  - CHAR(*n*) = fixed-length string of *n* characters.
  - VARCHAR(*n*) = variable-length string of up to *n* characters.



DDL - Primitive Types

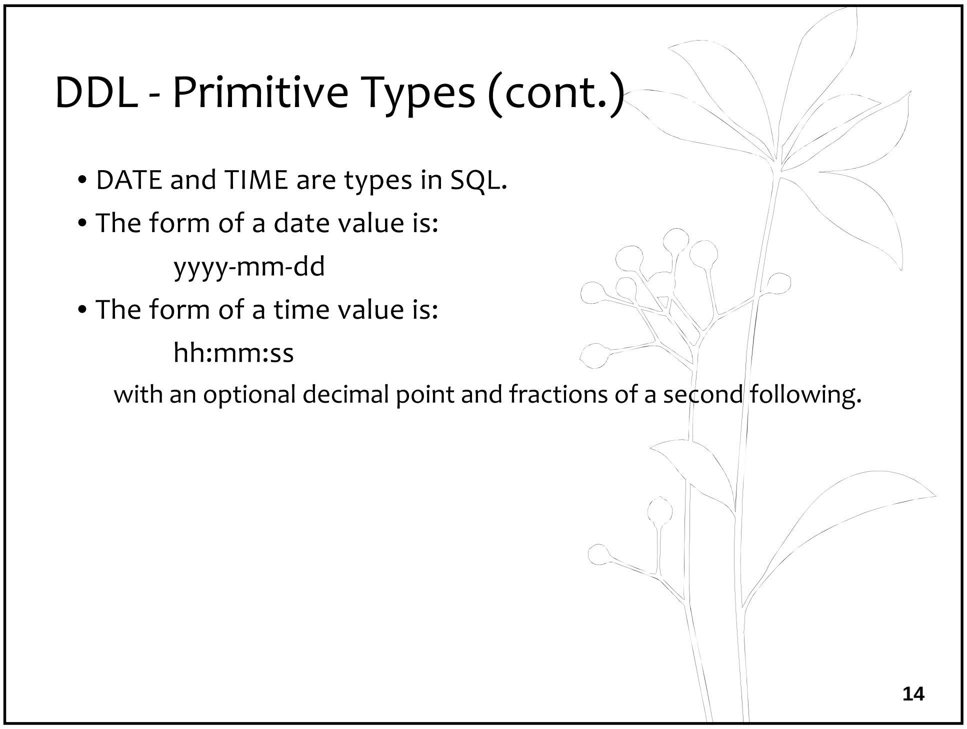
- numeric
  - INTEGER (or INT), SMALLINT
  - REAL, DOUBLE PRECISION
  - FLOAT(N)
  - DECIMAL(P,D) (or DEC(P,D), or NUMERIC(P,D))



DDL - Primitive Types (cont.)

- character-string
  - CHAR(N) (or CHARACTER(N))
  - VARCHAR(N) (or CHAR VARYING(N), or CHARACTER VARYING(N))
- bit-strings
  - BIT(N)
  - VARBIT(N) (or BIT VARYING(N))

13



DDL - Primitive Types (cont.)

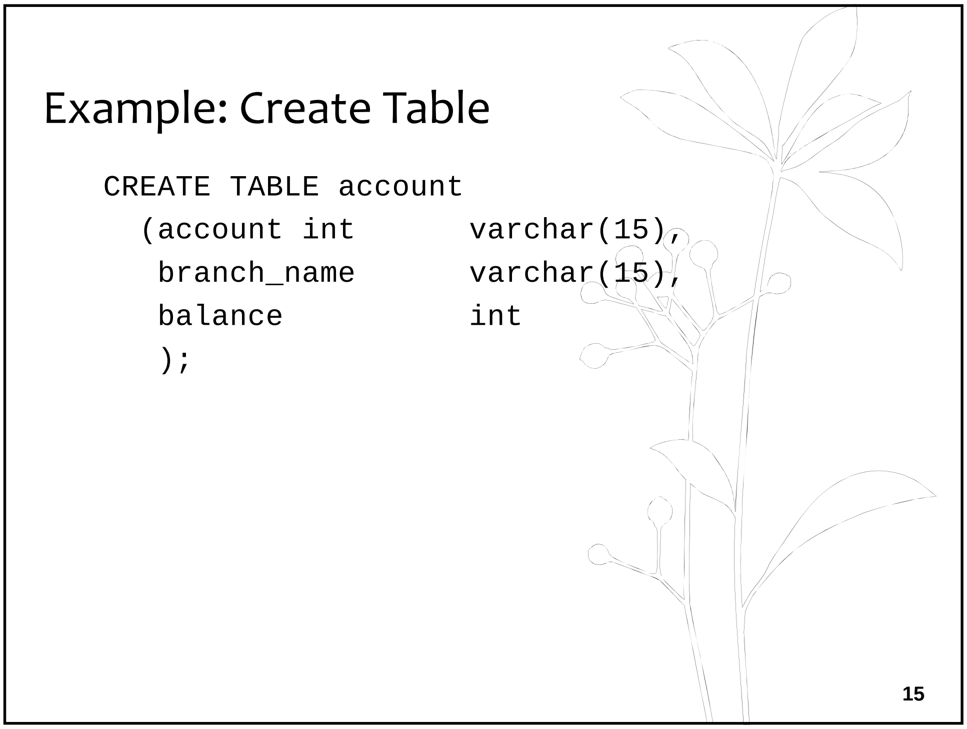
- DATE and TIME are types in SQL.
- The form of a date value is:

  yyyy-mm-dd
- The form of a time value is:

  hh:mm:ss

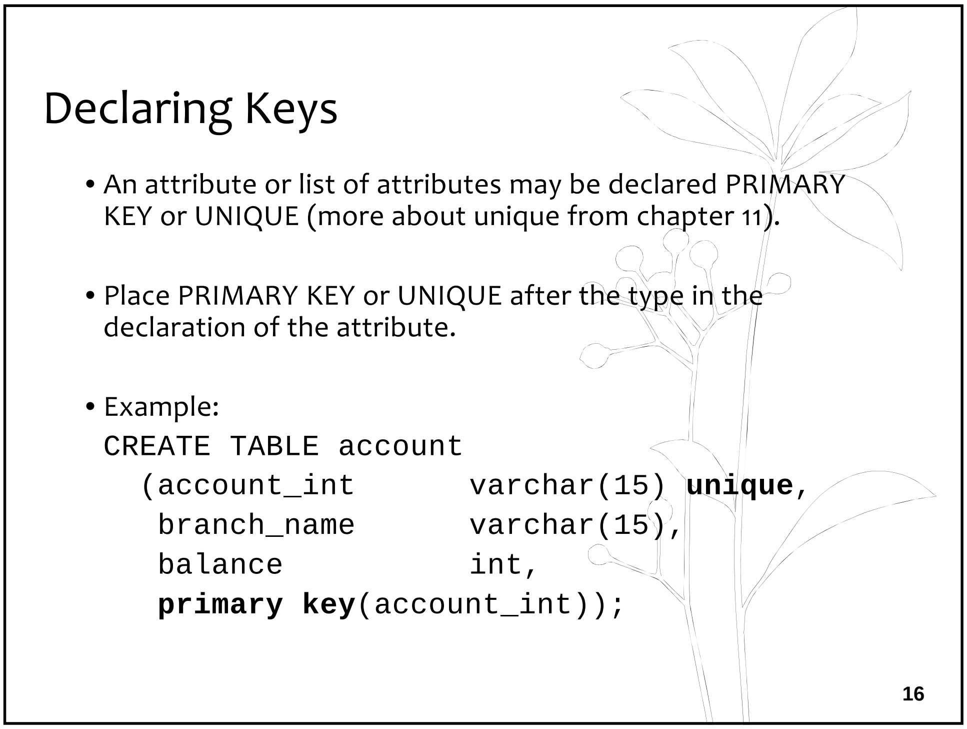
  with an optional decimal point and fractions of a second following.

14



Example: Create Table

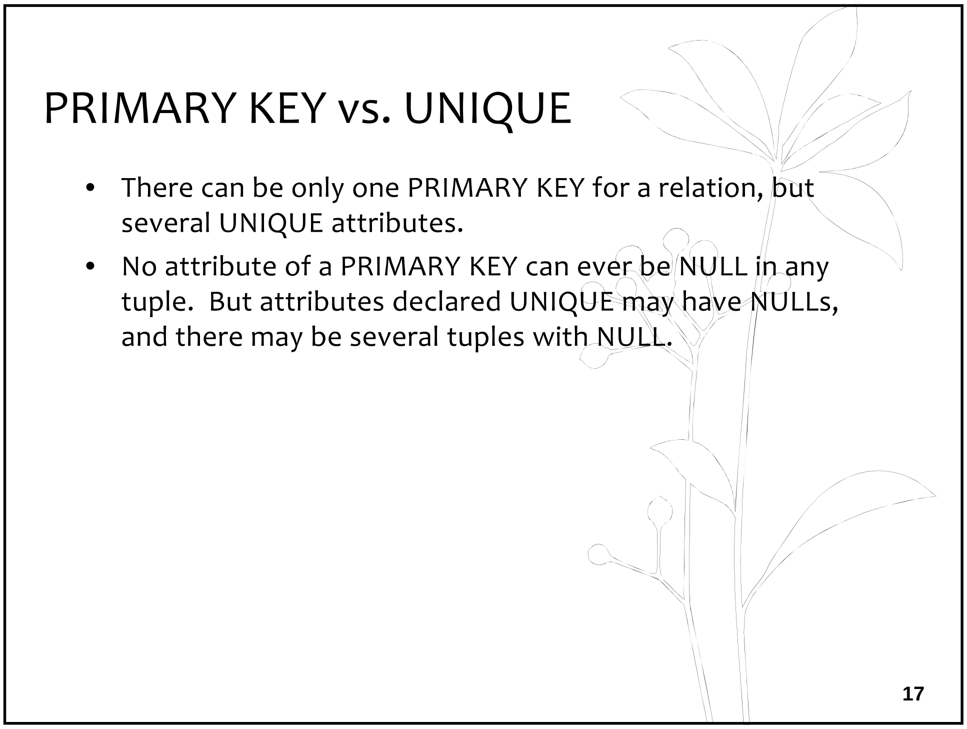
```
CREATE TABLE account
  (account int      varchar(15),
   branch_name      varchar(15),
   balance          int
   );
```



Declaring Keys

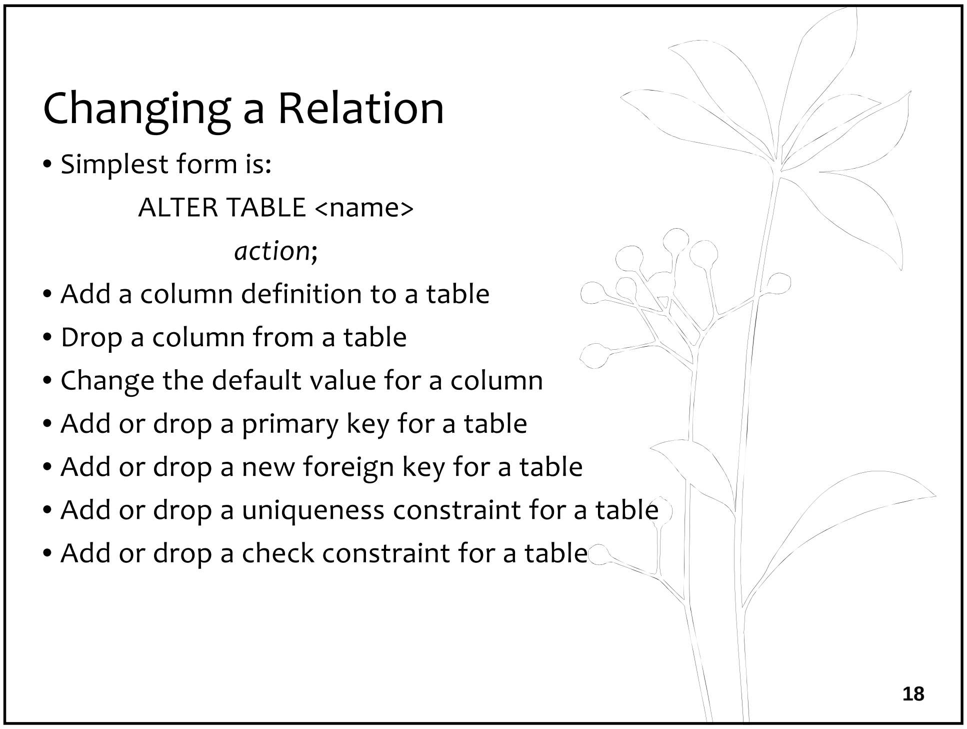
- An attribute or list of attributes may be declared PRIMARY KEY or UNIQUE (more about unique from chapter 11).
- Place PRIMARY KEY or UNIQUE after the type in the declaration of the attribute.
- Example:

```
CREATE TABLE account
  (account_int      varchar(15) unique,
   branch_name      varchar(15),
   balance          int,
   primary key(account_int));
```



PRIMARY KEY vs. UNIQUE

- There can be only one PRIMARY KEY for a relation, but several UNIQUE attributes.
- No attribute of a PRIMARY KEY can ever be NULL in any tuple. But attributes declared UNIQUE may have NULLs, and there may be several tuples with NULL.

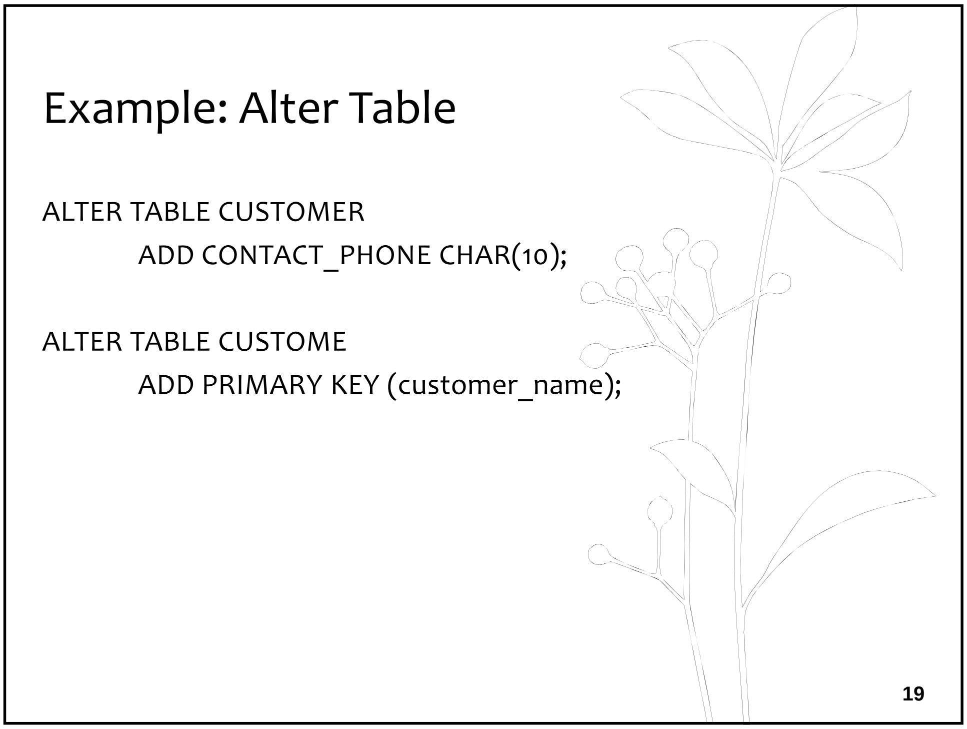


Changing a Relation

- Simplest form is:

```
ALTER TABLE <name>
        action;
```

- Add a column definition to a table
- Drop a column from a table
- Change the default value for a column
- Add or drop a primary key for a table
- Add or drop a new foreign key for a table
- Add or drop a uniqueness constraint for a table
- Add or drop a check constraint for a table



Example: Alter Table

```
ALTER TABLE CUSTOMER
        ADD CONTACT_PHONE CHAR(10);

ALTER TABLE CUSTOME
        ADD PRIMARY KEY (customer_name);
```