



# Chapter 3: Introduction to SQL

**Database System Concepts, 6<sup>th</sup> Ed.**

©Silberschatz, Korth and Sudarshan

See [www.db-book.com](http://www.db-book.com) for conditions on re-use



# Example of a Relation

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

attributes  
(or columns)

tuples  
(or rows)



# Attribute Types

- The set of allowed values for each attribute is called the **domain** of the attribute
- Attribute values are (normally) required to be **atomic**; that is, indivisible
- The special value ***null*** is a member of every domain
- The null value causes complications in the definition of many operations



# Data Definition Language

The Structured Query Language (SQL) **data-definition language (DDL)** allows the specification of information about relations, including:

- The schema for each relation.
- The domain of values associated with each attribute.
- Integrity constraints
- And as we will see later, also other information such as
  - The set of indices to be maintained for each relations.
  - Security and authorization information for each relation.
  - The physical storage structure of each relation on disk.



# Domain Types in SQL

- ❑ **char(*n*)**. Fixed length character string, with user-specified length *n*.
- ❑ **varchar(*n*)**. Variable length character strings, with user-specified maximum length *n*.
- ❑ **int**. Integer (a finite subset of the integers that is machine-dependent).
- ❑ **smallint**. Small integer (a machine-dependent subset of the integer domain type).
- ❑ **numeric(*p*,*n*)**. Fixed point number, with user-specified precision of *p* digits, with *n* digits to the right of decimal point.
- ❑ **real, double precision**. Floating point and double-precision floating point numbers, with machine-dependent precision.
- ❑ **float(*n*)**. Floating point number, with user-specified precision of at least *n* digits.
- ❑ More are covered in Chapter 4.



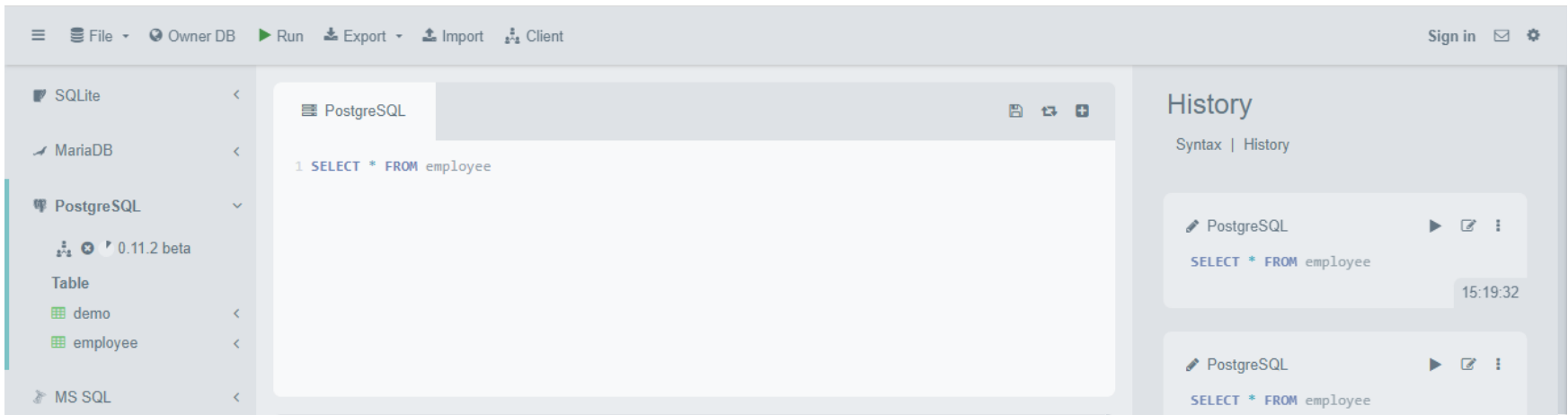
# Types of SQL statements

Type of SQL statement	SQL keywords	Function
Data Definition Language (DDL)	CREATE ALTER DROP	Used to define, change and drop the structure of a table.
	TRUNCATE	Used to remove all rows from a table.
Data Manipulation Language(DML)	INSERT INTO UPDATE DELETE FROM SELECT	Used to enter, modify, delete and retrieve data from a table.
Data Control Language (DCL)	GRANT REVOKE	Used to control access to the data in a database.
	COMMIT ROLLBACK	Used to define the end of a transaction.



# Online SQL Editor

<https://sqliteonline.com/>





# Create Table Construct

- An SQL relation is defined using the **create table** command:

```
create table  $r$  ( $A_1$   $D_1$ ,  $A_2$   $D_2$ , ...,  $A_n$   $D_n$ ,  
                (integrity-constraint1),  
                ...,  
                (integrity-constraintk))
```

- $r$  is the name of the relation
- each  $A_i$  is an attribute name in the schema of relation  $r$
- $D_i$  is the data type of values in the domain of attribute  $A_i$
- Example:

```
create table instructor (  
    ID           char(5),  
    name        varchar(20) not null,  
    dept_name varchar(20),  
    salary     numeric(8,2))
```

- **insert into** *instructor* **values** ('10211', 'Smith', 'Biology', 66000);
- **insert into** *instructor* **values** ('10211', null, 'Biology', 66000);





# Integrity Constraints in Create Table

- **not null**
- **primary key** ( $A_1, \dots, A_n$ )
- **foreign key** ( $A_m, \dots, A_n$ ) **references**  $r$

Example: Declare *dept\_name* as the primary key for *department*

.

```
create table instructor (  
    ID          char(5),  
    name        varchar(20) not null,  
    dept_name    varchar(20),  
    salary       numeric(8,2),  
    primary key (ID),  
    foreign key (dept_name) references department)
```

**primary key** declaration on an attribute automatically ensures **not null**



# And a Few More Relation Definitions

- **create table** *student* (  
    *ID*                **varchar**(5),  
    *name*            **varchar**(20) not null,  
    *dept\_name*       **varchar**(20),  
    *tot\_cred*        **numeric**(3,0),  
    **primary key** (*ID*),  
    **foreign key** (*dept\_name*) **references** *department*) );
- **create table** *takes* (  
    *ID*                **varchar**(5),  
    *course\_id*       **varchar**(8),  
    *sec\_id*           **varchar**(8),  
    *semester*        **varchar**(6),  
    *year*             **numeric**(4,0),  
    *grade*            **varchar**(2),  
    **primary key** (*ID*, *course\_id*, *sec\_id*, *semester*, *year*),  
    **foreign key** (*ID*) **references** *student*,  
    **foreign key** (*course\_id*, *sec\_id*, *semester*, *year*) **references** *section* );
- Note: *sec\_id* can be dropped from primary key above, to ensure a student cannot be registered for two sections of the same course in the same semester



# And more still

- ❑ **create table** *course* (  
    *course\_id*      **varchar(8) primary key**,  
    *title*          **varchar(50)**,  
    *dept\_name*     **varchar(20)**,  
    *credits*        **numeric(2,0)**,  
    **foreign key** (*dept\_name*) **references** *department*) );
- ❑ Primary key declaration can be combined with attribute declaration as shown above



# Drop and Alter Table Constructs

- **drop table** *student*
  - Deletes the table and its contents
- **delete from** *student*
  - Deletes all contents of table, but retains table
- **alter table**
  - **alter table** *r* **add** *A D*
    - ▶ where *A* is the name of the attribute to be added to relation *r* and *D* is the domain of *A*.
    - ▶ All tuples in the relation are assigned *null* as the value for the new attribute.
  - **alter table** *r* **drop** *A*
    - ▶ where *A* is the name of an attribute of relation *r*
    - ▶ Dropping of attributes not supported by many databases



# Basic Query Structure

- The SQL **data-manipulation language (DML)** provides the ability to query information, and insert, delete and update tuples
- A typical SQL query has the form:

**select**  $A_1, A_2, \dots, A_n$   
**from**  $r_1, r_2, \dots, r_m$   
**where**  $P$

- $A_i$  represents an attribute
  - $R_i$  represents a relation
  - $P$  is a predicate.
- The result of an SQL query is a relation.



# The select Clause

- The **select** clause list the attributes desired in the result of a query
  - corresponds to the projection operation of the relational algebra

- Example: find the names of all instructors:

```
select name  
from instructor
```

- NOTE: SQL names are case insensitive (i.e., you may use upper- or lower-case letters.)
  - E.g. *Name*  $\equiv$  *NAME*  $\equiv$  *name*
  - Some people use upper case wherever we use bold font.