# Intro to Databases

**Dr. Abdallah Mohamed**

Acknowledgement: Original slides provided courtesy of Dr. Lawrence.

## Key Points

1) Databases allow for easy storage and retrieval of large amounts of information.

2) Relational databases organize data into tables consisting of rows and columns.

3) SQL is the common language to query a database for results.



## What is a database?

A ***database*** is a collection of logically related data for a particular domain.

A ***database management system*** (***DBMS***) is software designed for the creation and management of databases.

- e.g. Oracle, DB2, Microsoft Access, MySQL, SQL Server

Bottom line: A ***database*** is the *data* stored and a ***database system*** is the *software* that manages the data.



## Databases in the Real-World

Databases are everywhere. Examples:

- Online web sites such as Amazon, eBay, and Expedia track orders, shipments, and customers using databases.
  - ⇨ eBay, with 5 petabytes (*Computerworld, Oct 14, 2008*)
- Retailers manage their products and sales using a database.
  - ⇨ Wal-Mart, with 2.5 petabyte. (Computerworld, Oct 14, 2008)
  - ⇨ Wal-Mart: Daily data from 800 million transactions by 30 million customers
- The university maintains all your registration information and marks in a database.

Can you think of other examples? What data do ***you*** have?



## DBMS

A database management system provides *efficient*, *convenient*, and *safe multi-user* storage and access to *massive* amounts of *persistent* data.

***Efficient*** - Able to handle large data sets and complex queries without searching all files and data items.

***Convenient*** - Easy to write queries to retrieve data.

***Safe*** - Protects data from system failures and hackers.

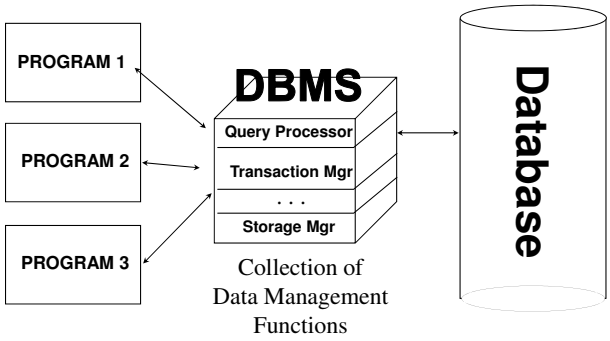
***Massive*** - Database sizes in gigabytes and terabytes.

***Persistent*** - Data exists even if have a power failure.

***Multi-user*** - More than one user can access and update data at the same time while preserving consistency.



## Database System Approach



# The Relational Model: Terminology

The ***relational model*** organizes database information into tables called **relations**.

- The relational model was developed by E. F. Codd in 1970 and is used by almost all commercial database systems.

Terminology:

A ***relation*** is a table with columns and rows.

An ***attribute*** is a named column of a relation.

A ***tuple*** is a row of a relation.

A ***domain*** is a set of allowable values for one or more attributes.

The ***degree*** of a relation is the number of attributes it contains.

The ***cardinality*** of a relation is the number of tuples it contains.



# Relation Example

*Employee*
**Relation**

**Attributes**

| Id | name | salary | Province |
|---|---|---|---|
| 2134 | Amy | 61000 | BC |
| 2137 | John | 62000 | AB |
| 3124 | Lee | 71400 | NL |
| 3234 | Lili | 51700 | AB |
| 1263 | Frank | 65000 | NL |

**Tuples**

**Degree = 4**
**Cardinality = 5**

**Domains of Employee Relation**
**Id – *integer***
***Name - string of alphabet characters***
**Salary - *currency*.**
**Province – set of provinces of Canada**



# Relation Example

**relation**

**attributes**

**tuples**

Products : Table

| Product ID | Product Name | Supplier | Category | Quantity Per Unit | Unit Price | Units In Stock |
|---|---|---|---|---|---|---|
| 1 | Chai | 1 | 1 | 10 boxes x 20 bags | $18.00 | 39 |
| 2 | Chang | 1 | 1 | 24 - 12 oz bottles | $19.00 | 17 |
| 3 | Aniseed Syrup | 1 | 2 | 12 - 550 ml bottles | $10.00 | 13 |
| 4 | Chef Anton's Cajun Seasoning | 2 | 2 | 48 - 6 oz jars | $22.00 | 53 |
| 5 | Chef Anton's Gumbo Mix | 2 | 2 | 36 boxes | $21.35 | 0 |
| 6 | Grandma's Boysenberry Spread | 3 | 2 | 12 - 8 oz jars | $25.00 | 120 |
| 7 | Uncle Bob's Organic Dried Pears | 3 | 7 | 12 - 1 lb pkgs. | $30.00 | 15 |
| 8 | Northwoods Cranberry Sauce | 3 | 2 | 12 - 12 oz jars | $40.00 | 6 |
| 9 | Mishi Kobe Niku | 4 | 6 | 18 - 500 g pkgs. | $97.00 | 29 |
| 10 | Ikura | 4 | 8 | 12 - 200 ml jars | $31.00 | 31 |
| 11 | Queso Cabrales | 5 | 4 | 1 kg pkg. | $21.00 | 22 |

Record: 1 of 77

**Degree = 7**
**Cardinality = 77**

**Domain of Unit Price is *currency*.**



# Relation Practice Questions

Order : Select Query

| Order ID | Customer | Employee | Order Date | Shipped Date | Ship Via | Ship Name | Ship Address | Ship Postal Code |
|---|---|---|---|---|---|---|---|---|
| 10248 | VINET | 5 | 04-Aug-94 | 16-Aug-94 | 3 | Vins et alcools Chevalier | 59 rue de l'Abbaye | 51100 |
| 10249 | TOMSP | 6 | 05-Aug-94 | 10-Aug-94 | 1 | Toms Spezialitäten | Luisenstr. 48 | 44087 |
| 10250 | HANAR | 4 | 08-Aug-94 | 12-Aug-94 | 2 | Hanari Carnes | Rua do Paço, 67 | 05454-876 |
| 10251 | VICTE | 3 | 08-Aug-94 | 15-Aug-94 | 1 | Victuailles en stock | 2, rue du Commerce | 69004 |
| 10252 | SUPRD | 4 | 09-Aug-94 | 11-Aug-94 | 2 | Suprêmes délices | Boulevard Tirou, 255 | B-6000 |
| 10253 | HANAR | 3 | 10-Aug-94 | 16-Aug-94 | 2 | Hanari Carnes | Rua do Paço, 67 | 05454-876 |
| 10254 | CHOPS | 5 | 11-Aug-94 | 23-Aug-94 | 2 | Chop-suey Chinese | Hauptstr. 31 | 3012 |
| 10255 | RICSU | 9 | 12-Aug-94 | 15-Aug-94 | 3 | Richter Supermarkt | Starenweg 5 | 1204 |
| 10256 | WELLI | 3 | 15-Aug-94 | 17-Aug-94 | 2 | Wellington Importadora | Rua do Mercado, 12 | 08737-363 |
| 10257 | HILAA | 4 | 16-Aug-94 | 22-Aug-94 | 3 | HILARIÓN-Abastos | Carrera 22 con Ave. Carlos | 5022 |
| 10258 | ERNSH | 1 | 17-Aug-94 | 23-Aug-94 | 1 | Ernst Handel | Kirchgasse 6 | 8010 |
| 10259 | CENTC | 4 | 18-Aug-94 | 25-Aug-94 | 3 | Centro comercial Moctezuma | Sierras de Granada 9993 | 05022 |
| 10260 | OTTIK | 4 | 19-Aug-94 | 29-Aug-94 | 1 | Ottilies Käseladen | Mehrheimerstr. 369 | 50739 |

Record: 1 of 827

1) What is the name of the relation?

2) What is the cardinality of the relation?

3) What is the degree of the relation?

4) What is the domain of order date? What is the domain of order id?



# Databases
# Database and Database System

***Question:*** Which of these two definitions below are an example of software?

**A)** database

**B)** database system



# Databases
# Database Properties

***Question:*** True or False: The data in a database is lost when the power to the computer is turned off.

**A)** true

**B)** false

## Databases
## Database Properties (2)

***Question:*** True or False: More than one user can use the database managed by the DBMS at the same time.

**A)** true

**B)** false



## Databases
## Definition Matching

***Question:*** Given the three definitions, select the ordering that contains their related definitions.

Relation, Tuple, Attribute

**A)** column, row, table
**B)** row, column, table
**C)** table, row, column
**D)** table, column, row



## Databases
## Cardinality and Degree

***Question:*** A database table has 10 rows and 5 columns. Select ***one*** true statement.

**A)** The table's degree is 50.

**B)** The table's cardinality is 5.

**C)** The table's degree is 10.

**D)** The table's cardinality is 10.



## Relational Keys

Keys are used to ***uniquely identify*** a tuple in a relation.

A ***key*** is a *minimal* set of attributes that uniquely identifies a tuple in a relation.

- We will consider only the case where the key is represented by ONE attribute

There is also another term called ***superkey***, but we will not discuss it here.



## Databases
## Keys (2)

***Question:*** True or false: It is possible to have more than one key for a table.

**A)** true

**B)** false



## Example Relations

Relations:

**emp** (**eno**, ename, bdate, title, salary, supereno, dno)
**proj** (**pno**, pname, budget, dno)
**dept** (**dno**, dname, mgreno)
**workson** (eno, pno, resp, hours) ← We will skip this table for now

`Emp` - one row per employee storing name, birth date, supervisor, and department that they are in

`Proj` - one row per project storing name and its department

`Dept` - one row per department storing name and manager

`WorksOn` - stores that an employee works on a particular project for a certain amount of time in a given role

Note: Key fields are underlined.

## Example Relation Instances

We will skip this table for now

Emp Relation

| eno | ename | bdate | title | salary | supereno | dno |
|---|---|---|---|---|---|---|
| E1 | J. Doe | 01-05-75 | EE | 30000 | E2 | null |
| E2 | M. Smith | 06-04-66 | SA | 50000 | E5 | D3 |
| E3 | A. Lee | 07-05-66 | ME | 40000 | E7 | D2 |
| E4 | J. Miller | 09-01-50 | PR | 20000 | E6 | D3 |
| E5 | B. Casey | 12-25-71 | SA | 50000 | E8 | D3 |
| E6 | L. Chu | 11-30-65 | EE | 30000 | E7 | D2 |
| E7 | R. Davis | 09-08-77 | ME | 40000 | E8 | D1 |
| E8 | J. Jones | 10-11-72 | SA | 50000 | null | D1 |

WorksOn Relation

| eno | pno | resp | hours |
|---|---|---|---|
| E1 | P1 | Manager | 12 |
| E2 | P1 | Analyst | 24 |
| E2 | P2 | Analyst | 6 |
| E3 | P3 | Consultant | 10 |
| E3 | P4 | Engineer | 48 |
| E4 | P2 | Programmer | 18 |
| E5 | P2 | Manager | 24 |
| E6 | P4 | Manager | 48 |
| E7 | P3 | Engineer | 36 |

Proj Relation

| pno | pname | budget | dno |
|---|---|---|---|
| P1 | Instruments | 150000 | D1 |
| P2 | DB Develop | 135000 | D2 |
| P3 | Budget | 250000 | D3 |
| P4 | Maintenance | 310000 | D2 |
| P5 | CAD/CAM | 500000 | D2 |

Dept Relation

| dno | dname | mgreno |
|---|---|---|
| D1 | Management | E8 |
| D2 | Consulting | E7 |
| D3 | Accounting | E5 |
| D4 | Development | null |



## Practice

Look at the tables in the previous slide and try to answer these questions:

1. What is the salary of Miller?
2. Who is the supervisor of Miller?
3. What is the department name where Miller works?
4. Who is the manager of the department of Miller?
5. What is the salary of the manger of the Consulting dept.?
6. What is the total budget of the projects in the department managed by Davis?



## How to search DB?



## SQL Overview

Structured Query Language or SQL is the standard database query language to retrieve *exact answers*.

- SQL is a ***declarative language*** (non-procedural).
  - ⇨A SQL query specifies ***WHAT*** to retrieve but ***not HOW*** to retrieve it.
- SQL is used by Microsoft Access.

Some basic rules for SQL statements:

- 1) There is a set of *reserved words* that cannot be used as names for database fields and tables.
  - ⇨`SELECT`, `FROM`, `WHERE`, etc.
- 2) SQL is generally *case-insensitive*.
  - ⇨Only exception is string constants. 'FRED' not the same as 'fred'.
- 3) SQL is *free-format* and white-space is ignored.



## SQL Queries

A query in SQL has the form:

**SELECT (list of attributes)**
**FROM (list of tables)**
**WHERE (filter *conditions*)**

Notes:

- 1) Separate the list of attributes and list of tables by **commas**.
- 2) The "`*`" is used to select all attributes.



## SQL
## Retrieving Only Some of the Columns

The ***projection operation*** creates a new table that has some of the columns of the input table. In SQL, provide the table in the `FROM` clause and the fields in the output in the `SELECT`.

Example: Return only the `eno` field from the `Emp` table:

```
SELECT eno
FROM   emp
```

Emp Relation

| eno | ename | bdate | title | salary | supereno | dno |
|---|---|---|---|---|---|---|
| E1 | J. Doe | 01-05-75 | EE | 30000 | E2 | null |
| E2 | M. Smith | 06-04-66 | SA | 50000 | E5 | D3 |
| E3 | A. Lee | 07-05-66 | ME | 40000 | E7 | D2 |
| E4 | J. Miller | 09-01-50 | PR | 20000 | E6 | D3 |
| E5 | B. Casey | 12-25-71 | SA | 50000 | E8 | D3 |
| E6 | L. Chu | 11-30-65 | EE | 30000 | E7 | D2 |
| E7 | R. Davis | 09-08-77 | ME | 40000 | E8 | D1 |
| E8 | J. Jones | 10-11-72 | SA | 50000 | null | D1 |

→

Result

| eno |
|---|
| E1 |
| E2 |
| E3 |
| E4 |
| E5 |
| E6 |
| E7 |
| E8 |

## SQL Projection Examples

Emp Relation

| eno | ename | title | salary |
|---|---|---|---|
| E1 | J. Doe | EE | 30000 |
| E2 | M. Smith | SA | 50000 |
| E3 | A. Lee | ME | 40000 |
| E4 | J. Miller | PR | 20000 |
| E5 | B. Casey | SA | 50000 |
| E6 | L. Chu | EE | 30000 |
| E7 | R. Davis | ME | 40000 |
| E8 | J. Jones | SA | 50000 |

```
SELECT eno,ename
FROM   emp
```

| eno | ename |
|---|---|
| E1 | J. Doe |
| E2 | M. Smith |
| E3 | A. Lee |
| E4 | J. Miller |
| E5 | B. Casey |
| E6 | L. Chu |
| E7 | R. Davis |
| E8 | J. Jones |

```
SELECT title
FROM   emp
```

| title |
|---|
| EE |
| SA |
| ME |
| PR |
| SA |
| EE |
| ME |
| SA |

Note: Duplicates are not removed during SQL projection.



## Databases Projection

***Question:*** Given this table and the query:

```
SELECT eno, ename, salary
FROM   emp
```

How many columns are returned?

**A)** 0
**B)** 1
**C)** 2
**D)** 3
**E)** 4

Emp Relation

| eno | ename | title | salary |
|---|---|---|---|
| E1 | J. Doe | EE | 30000 |
| E2 | M. Smith | SA | 50000 |
| E3 | A. Lee | ME | 40000 |
| E4 | J. Miller | PR | 20000 |
| E5 | B. Casey | SA | 50000 |
| E6 | L. Chu | EE | 30000 |
| E7 | R. Davis | ME | 40000 |
| E8 | J. Jones | SA | 50000 |



## Databases Projection (2)

***Question:*** Given this table and the query:

```
SELECT salary
FROM   emp
```

How many rows are returned?

**A)** 0
**B)** 2
**C)** 4
**D)** 8

Emp Relation

| eno | ename | title | salary |
|---|---|---|---|
| E1 | J. Doe | EE | 30000 |
| E2 | M. Smith | SA | 50000 |
| E3 | A. Lee | ME | 40000 |
| E4 | J. Miller | PR | 20000 |
| E5 | B. Casey | SA | 50000 |
| E6 | L. Chu | EE | 30000 |
| E7 | R. Davis | ME | 40000 |
| E8 | J. Jones | SA | 50000 |



## SQL Projection Questions

WorksOn Relation

| eno | pno | resp | dur |
|---|---|---|---|
| E1 | P1 | Manager | 12 |
| E2 | P1 | Analyst | 24 |
| E2 | P2 | Analyst | 6 |
| E3 | P3 | Consultant | 10 |
| E3 | P4 | Engineer | 48 |
| E4 | P2 | Programmer | 18 |
| E5 | P2 | Manager | 24 |
| E6 | P4 | Manager | 48 |
| E7 | P3 | Engineer | 36 |
| E7 | P5 | Engineer | 23 |
| E8 | P3 | Manager | 40 |

Write the SQL statement that:
1) Returns only attributes *resp* and *dur*.
2) Returns only *eno*.
3) Returns only *pno*.

List the number of result rows and columns in each case.



## One Table Query Example Retrieving Only Some of the Rows

The ***selection operation*** creates a new table with some of the rows of the input table. A condition specifies which rows are in the new table. The condition is similar to an `if` statement.

Example: Return the projects in department `'D2'`:

```
SELECT pno, pname, budget, dno
FROM   proj
WHERE  dno = 'D2';
```

Proj Relation

| pno | pname | budget | dno |
|---|---|---|---|
| P1 | Instruments | 150000 | D1 |
| P2 | DB Develop | 135000 | D2 |
| P3 | Budget | 250000 | D3 |
| P4 | Maintenance | 310000 | D2 |
| P5 | CAD/CAM | 500000 | D2 |

→

Result

| pno | pname | budget | dno |
|---|---|---|---|
| P2 | DB Develop | 135000 | D2 |
| P4 | Maintenance | 310000 | D2 |
| P5 | CAD/CAM | 500000 | D2 |

Algorithm: Scan each tuple and check if matches condition in `WHERE` clause.



## Retrieving Only Some of the Rows Selection Conditions

The condition in a selection statement specifies which rows are included. It has the general form of an if statement.

The condition may consist of attributes, constants, comparison operators (`<`, `>`, `=`, `!=`, `<=`, `>=`), and logical operators (`AND`, `OR`, `NOT`).

## SQL Selection Examples

Emp Relation

| eno | ename | title | salary |
|---|---|---|---|
| E1 | J. Doe | EE | 30000 |
| E2 | M. Smith | SA | 50000 |
| E3 | A. Lee | ME | 40000 |
| E4 | J. Miller | PR | 20000 |
| E5 | B. Casey | SA | 50000 |
| E6 | L. Chu | EE | 30000 |
| E7 | R. Davis | ME | 40000 |
| E8 | J. Jones | SA | 50000 |

```
SELECT eno, ename, title, salary
FROM   emp
WHERE  salary > 35000 OR
       title = 'PR'
```

| eno | ename | title | salary |
|---|---|---|---|
| E2 | M. Smith | SA | 50000 |
| E3 | A. Lee | ME | 40000 |
| E4 | J. Miller | PR | 20000 |
| E5 | B. Casey | SA | 50000 |
| E7 | R. Davis | ME | 40000 |
| E8 | J. Jones | SA | 50000 |

```
SELECT *
FROM   emp
WHERE  title = 'EE'
```

| eno | ename | title | salary |
|---|---|---|---|
| E1 | J. Doe | EE | 30000 |
| E6 | L. Chu | EE | 30000 |



## Databases Selection

***Question:*** Given this table and the query:

```
SELECT *
FROM   emp
WHERE  title='EE'
```

How many rows are returned?

**A)** 0
**B)** 1
**C)** 2
**D)** 3

Emp Relation

| eno | ename | title | salary |
|---|---|---|---|
| E1 | J. Doe | EE | 30000 |
| E2 | M. Smith | SA | 50000 |
| E3 | A. Lee | ME | 40000 |
| E4 | J. Miller | PR | 20000 |
| E5 | B. Casey | SA | 50000 |
| E6 | L. Chu | EE | 30000 |
| E7 | R. Davis | ME | 40000 |
| E8 | J. Jones | SA | 50000 |



## Databases Selection

***Question:*** Given this table and the query:

```
SELECT *
FROM   emp
WHERE  salary > 50000 or title='PR'
```

How many rows are returned?

**A)** 0
**B)** 1
**C)** 2
**D)** 3

Emp Relation

| eno | ename | title | salary |
|---|---|---|---|
| E1 | J. Doe | EE | 30000 |
| E2 | M. Smith | SA | 50000 |
| E3 | A. Lee | ME | 40000 |
| E4 | J. Miller | PR | 20000 |
| E5 | B. Casey | SA | 50000 |
| E6 | L. Chu | EE | 30000 |
| E7 | R. Davis | ME | 40000 |
| E8 | J. Jones | SA | 50000 |



## Databases Selection

***Question:*** Given this table and the query:

```
SELECT *
FROM   emp
WHERE  salary > 50000 or title='PR'
```

How many columns are returned?

**A)** 0
**B)** 2
**C)** 3
**D)** 4

Emp Relation

| eno | ename | title | salary |
|---|---|---|---|
| E1 | J. Doe | EE | 30000 |
| E2 | M. Smith | SA | 50000 |
| E3 | A. Lee | ME | 40000 |
| E4 | J. Miller | PR | 20000 |
| E5 | B. Casey | SA | 50000 |
| E6 | L. Chu | EE | 30000 |
| E7 | R. Davis | ME | 40000 |
| E8 | J. Jones | SA | 50000 |



## SQL Selection Questions

WorksOn Relation

| eno | pno | resp | dur |
|---|---|---|---|
| E1 | P1 | Manager | 12 |
| E2 | P1 | Analyst | 24 |
| E2 | P2 | Analyst | 6 |
| E3 | P3 | Consultant | 10 |
| E3 | P4 | Engineer | 48 |
| E4 | P2 | Programmer | 18 |
| E5 | P2 | Manager | 24 |
| E6 | P4 | Manager | 48 |
| E7 | P3 | Engineer | 36 |
| E7 | P5 | Engineer | 23 |
| E8 | P3 | Manager | 40 |

Write the SQL statement that:

1) Returns all rows with a project `P2`.
2) Returns all rows with responsibility of a `Manager`.
3) Returns all rows with a responsibility of `Manager` **and** duration of more than `40` months.

List the number of result rows for each case.



## One Table Query Example Retrieving Some of the Rows/Columns

Return the employee name and salary of all employees whose title is 'EE':

```
SELECT ename, salary
FROM   emp
WHERE  title = 'EE';
```

Emp Relation

| eno | ename | bdate | title | salary | supereno | dno |
|---|---|---|---|---|---|---|
| E1 | J. Doe | 01-05-75 | EE | 30000 | E2 | null |
| E2 | M. Smith | 06-04-66 | SA | 50000 | E5 | D3 |
| E3 | A. Lee | 07-05-66 | ME | 40000 | E7 | D2 |
| E4 | J. Miller | 09-01-50 | PR | 20000 | E6 | D3 |
| E5 | B. Casey | 12-25-71 | SA | 50000 | E8 | D3 |
| E6 | L. Chu | 11-30-65 | EE | 30000 | E7 | D2 |
| E7 | R. Davis | 09-08-77 | ME | 40000 | E8 | D1 |
| E8 | J. Jones | 10-11-72 | SA | 50000 | null | D1 |



Result

| ename | salary |
|---|---|
| J. Doe | 30000 |
| L. Chu | 30000 |

## One Table Query Examples

Return the birth date and salary of employee 'J. Doe':

```
SELECT bdate, salary
FROM   emp
WHERE  ename = 'J. Doe'
```

Return all information on all employees:

```
SELECT *          ← * returns all attributes
FROM   emp
```

Return the employee number, project number, and number of hours worked where the hours worked is > 50:

```
SELECT eno, pno, hours
FROM   workson
WHERE  hours > 50
```



## Databases
## Projection and Selection

***Question:*** Given this table and the query:

```
SELECT eno, salary
FROM   emp
WHERE  salary >= 40000
```

What is the degree of the result?

**A)** 2
**B)** 3
**C)** 4
**D)** 5

Emp Relation

| eno | ename | title | salary |
|---|---|---|---|
| E1 | J. Doe | EE | 30000 |
| E2 | M. Smith | SA | 50000 |
| E3 | A. Lee | ME | 40000 |
| E4 | J. Miller | PR | 20000 |
| E5 | B. Casey | SA | 50000 |
| E6 | L. Chu | EE | 30000 |
| E7 | R. Davis | ME | 40000 |
| E8 | J. Jones | SA | 50000 |



## Databases
## Projection and Selection (2)

***Question:*** Given this table and the query:

```
SELECT eno, salary
FROM   emp
WHERE  salary >= 40000
```

What is the cardinality of the result?

**A)** 2
**B)** 3
**C)** 4
**D)** 5

Emp Relation

| eno | ename | title | salary |
|---|---|---|---|
| E1 | J. Doe | EE | 30000 |
| E2 | M. Smith | SA | 50000 |
| E3 | A. Lee | ME | 40000 |
| E4 | J. Miller | PR | 20000 |
| E5 | B. Casey | SA | 50000 |
| E6 | L. Chu | EE | 30000 |
| E7 | R. Davis | ME | 40000 |
| E8 | J. Jones | SA | 50000 |



## SQL Projection/Selection
## One Table Questions

Relations:

emp (eno, ename, bdate, title, salary, supereno, dno)
proj (pno, pname, budget, dno)
dept (dno, dname, mgreno)
workson (eno, pno, resp, hours)

1) Returns all `employees` making more than $`50,000`.

2) Show the `WorksOn` records with less than `20` hours but more than `10` hours.

3) Return only the `pno` and `dno` for each `project`.

4) Return the name for each employee in department `'D1'`.

5) **Challenge:** Display the employees who (make less than $`40,000` or have title `'EE'`) and are born after June 1, 1970.

- Dates are in `YYYY-MM-DD` format. e.g. `#1970-06-01#`



## Putting it All Together

The steps to write an English query in SQL are:

- 1) Find the columns that you need and put in `SELECT` clause.
- 2) List the table that has the columns in the `FROM` clause.
- 3) If you must filter rows, add a filter criteria in `WHERE` clause.



## More Advanced Querying

There are many more queries that we can ask a database:

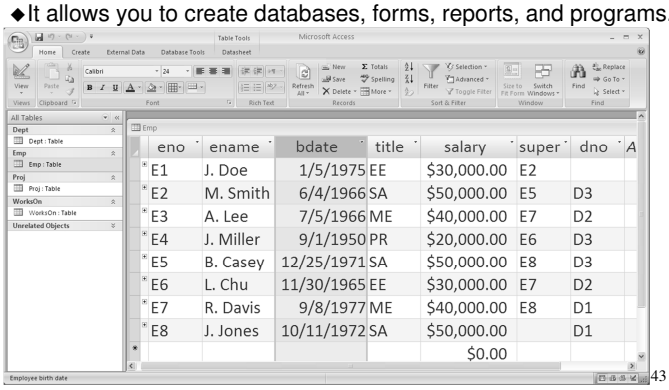
- querying data from MORE THAN ONE tables
- ordering results
- compute expressions and functions
- group data by value and meaning
- compute summary (aggregate) functions (max, min, sum, etc.)
- subqueries (queries within queries)

We will not study the notation for this advanced querying.

## Microsoft Access

Microsoft Access is a simple database management system.

- It allows you to create databases, forms, reports, and programs.

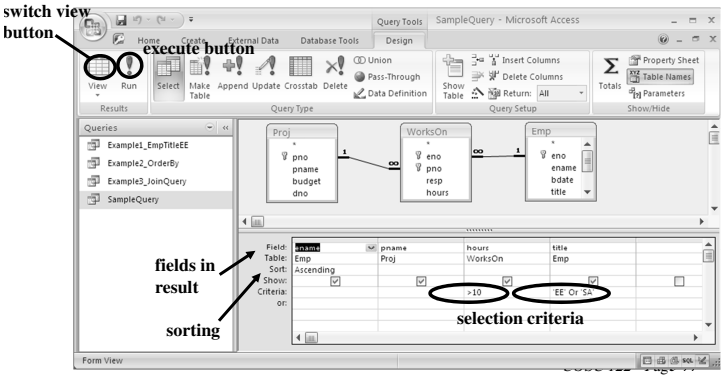


| eno | ename | bdate | title | salary | super | dno |
|---|---|---|---|---|---|---|
| E1 | J. Doe | 1/5/1975 | EE | $30,000.00 | E2 | |
| E2 | M. Smith | 6/4/1966 | SA | $50,000.00 | E5 | D3 |
| E3 | A. Lee | 7/5/1966 | ME | $40,000.00 | E7 | D2 |
| E4 | J. Miller | 9/1/1950 | PR | $20,000.00 | E6 | D3 |
| E5 | B. Casey | 12/25/1971 | SA | $50,000.00 | E8 | D3 |
| E6 | L. Chu | 11/30/1965 | EE | $30,000.00 | E7 | D2 |
| E7 | R. Davis | 9/8/1977 | ME | $40,000.00 | E8 | D1 |
| E8 | J. Jones | 10/11/1972 | SA | $50,000.00 | | D1 |
| | | | | $0.00 | | |



## Microsoft Access Query Interface

Tables are boxes. Relationships are lines. Condition specified on bottom.




## Microsoft Access Querying Basics

1) Projection is performed by selecting the fields in the output in the field row in the table at the bottom of the screen.

2) Selection is performed by entering the condition in the criteria box. The criteria applies to the field in that column.

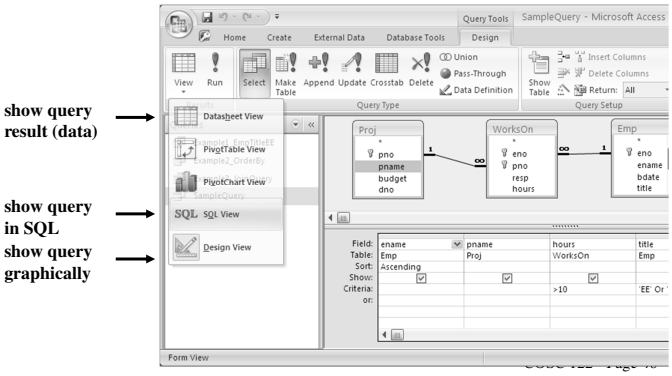
3) The tables used are added to the query by the **`Show Table…`** option.

4) Joins (based on relationships) are often automatically added, but if not, you can add them by selecting the join field in one table, holding the mouse button, then dragging to the join field in the other table.



## Microsoft Access Query Views

You may view your data, your query graphically, or your query in SQL.




## Conclusion

A ***database*** is a collection of related data. A ***database system*** allows storing and querying a database.

The basic query operations are selection (subset of rows), projection (subset of columns), and join (combine two or more tables).

***SQL*** is the standard query language for databases, although Microsoft Access also provides a graphical user interface.



## Objectives

- Define: database, database system
- Define: relation, attribute, tuple, domain, degree, cardinality, key
- Given a relation, know its cardinality, degree, domains, and keys.

Given a relational schema and instance be able to translate very simple English queries into SQL.