

# Management Information Systems

Larisa Cherkasov, MBA

Martin V. Smith School of Business and  
Economics

CSU Channel Islands

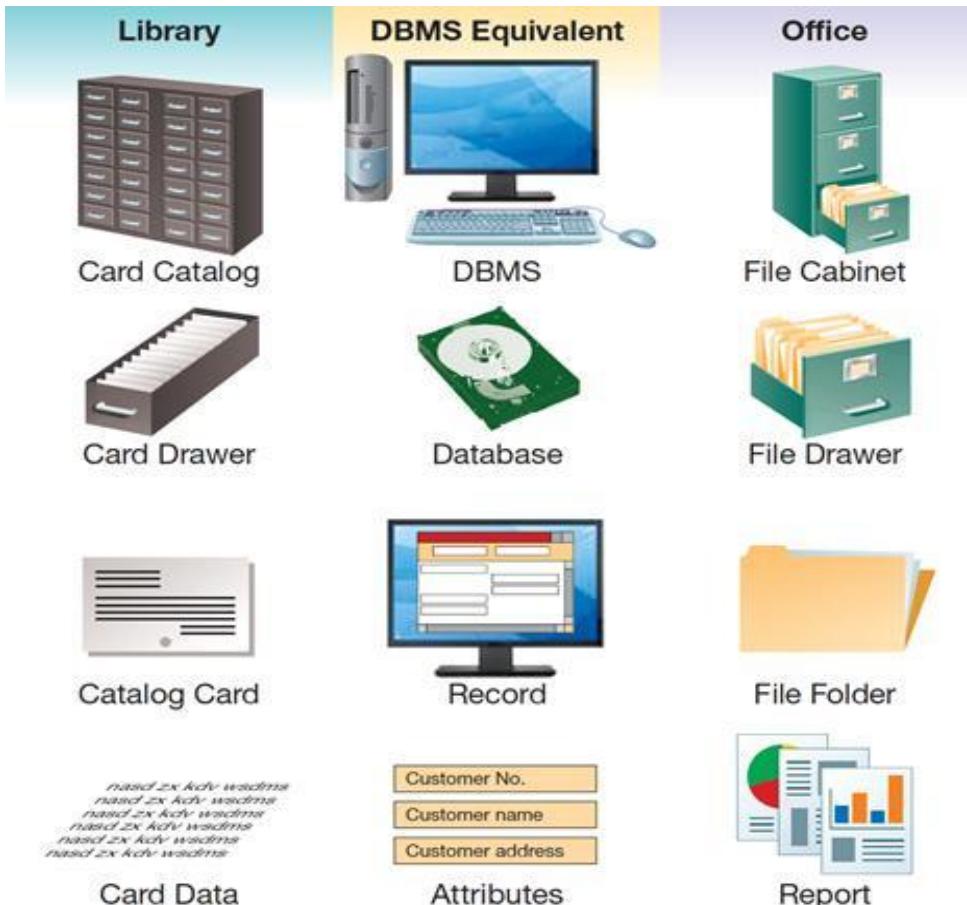
Email: [Larisa.Cherkasov@csuci.edu](mailto:Larisa.Cherkasov@csuci.edu)

# Relationship Databases

# Databases: Foundation Concepts



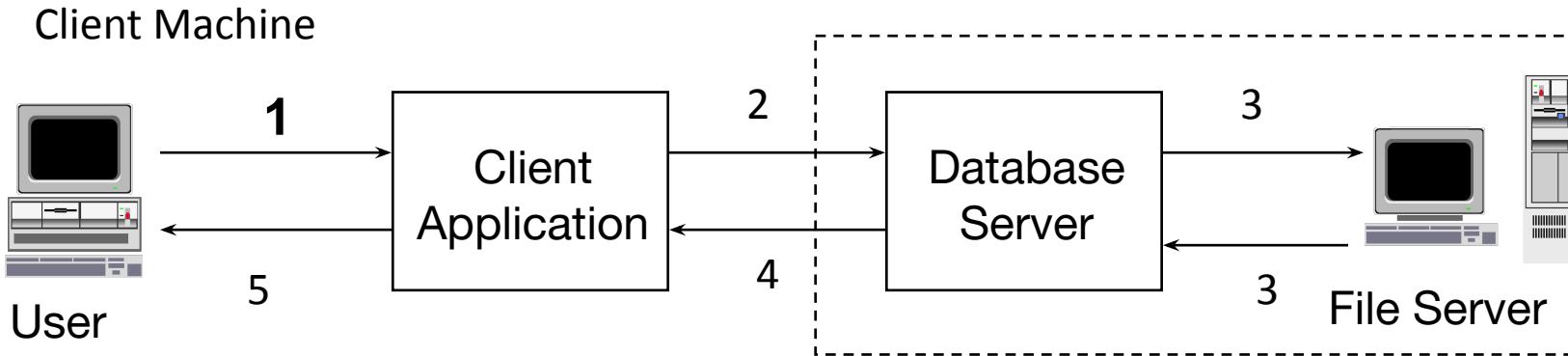
- **Database (DB):** Integrated collection of shared data designed to meet the varied information needs of an organization.



# Databases: Advantages

Advantages	Description
Minimal data redundancy	There is a single copy of the data
Improved data consistency	Eliminating redundancy improves consistency
Increase security	Simplifies enforcement of access restrictions
Improved data quality	Centralized control, minimized redundancy, and improved data consistency enhance data quality
Improved data accessibility and sharing	A centralized system makes it easier to deploy and control access in or out of the organization
Enforcement of standards	Centralized data management standardizes rules
Program-data independence	Easier to evolve and alter software to changing business needs when independent
Increased productivity of application development	Data standards make it easier to build and modify applications
Reduced program maintenance	Changes in data replicate seamlessly

# Databases: Client-Server Architecture



1. User enters query through client application's user interface.
2. Client application sends the formulated query to the database server.
3. Database server processes request. Search all records.
4. Database server returns only the results from the query.
5. Client application's user interface displays retrieved information.
  - Mechanism: SQL via SQL\*Net, ODBC, Stored Procedures, Remote Procedure Calls
  - Access is not a database server.

# Databases: Foundation Concepts

---

- **A Database Management System (DBMS):** Collection of software program that receives and satisfies all requests for data.
- **A Relational Database Management System ( RDBMS):** DBMS which processes data with the Relational Data Model.
  - **Relational databases**
    - Information is organized into related tables.
    - Each table captures information about a different entity.
    - Columns are different fields of information
      - attributes of the entity
    - Each row represents one instance
      - a specific example of the entity
    - The most prevalent type of database used.

# Databases: Benefits of RDBMS

---



- Integration of Data
  - Data are organized into a single, logical structure
- Sharing of Data
- Enforcement of Standards
  - Centralized data administration function
  - All changes to the data standards have to be approved by the Database Administrator.

# Databases: Benefits of RDBMS

---



- Uniform Security, Privacy and Integrity Controls
  - Controls for accessing, updating and protecting data are established by data administration
- Data Accessibility and Responsiveness
  - Provides multiple retrieval paths to each item of data
- Data Independence
  - The separation of data descriptions from application programs that use the data.

# Databases: Benefits of RDBMS

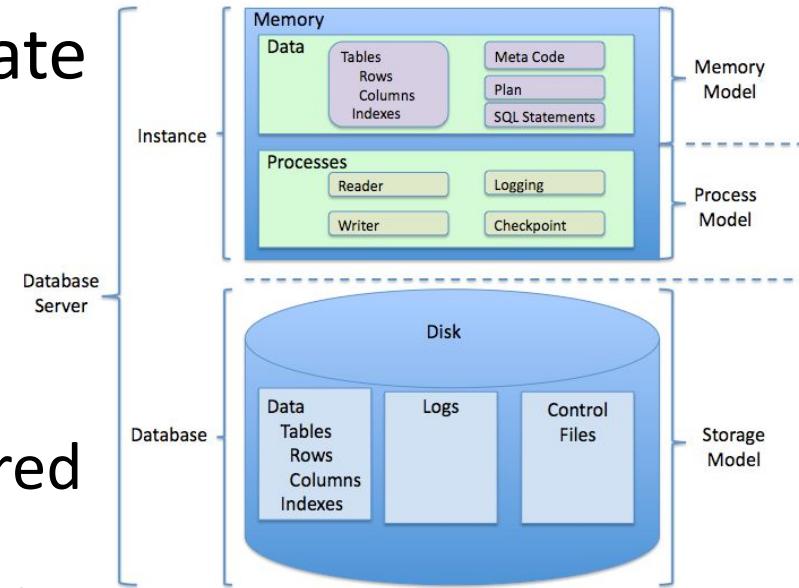
---



- Reduces Program Maintenance
  - Due to the data independence
- User Friendly
  - SQL is easy-to-learn language that allows to work with higher level data structures
- Supported by Mathematics Theory
  - RDBMS is the only DBMS that is supported by the relational algebra

# Databases: Relationship Model

- Invented by E. F. Codd in 1969
- Dominate the markets since late 1980s
- Strengths:
  - Simplicity
  - End-user orientation
  - Standardization (SQL – Structured Query Language)
  - Value-based instead of pointer-based to provide data independence
  - Endorsed by major computer companies



[Relationship Database Basics](#)

|

# Databases: Relationship Model

---



- A model represents both entities and relationships in a table structure.
  - **Relation:** A mathematical concept about an entity
- Properties of a Relation:
  - No duplicate rows in a relation.
  - Each row must be uniquely identified by a primary key.
  - The order of rows in a relation is insignificant
  - The order of columns in a relation is insignificant.
  - All attributes must be atomic (single value).
- Data are logically presented in two dimensional tables (files)
  - rows (records)
  - columns (attributes, fields, data item).

# Databases: Relationship Model

---



- **Table:** Group of homogeneously defined records
  - Each table has a unique name.
  - Each table captures information about a different entity
- **Record:** Collection of values that describe an entity.
  - The rows of a table are known as records.
- **Field:** Contains a single piece of data for the subject of the record
  - The columns of a table are known as fields ( or attributes)
- **Key:** An attribute or a set of attributes which can uniquely identify a record in a table.

# Databases: Relationship Model

---

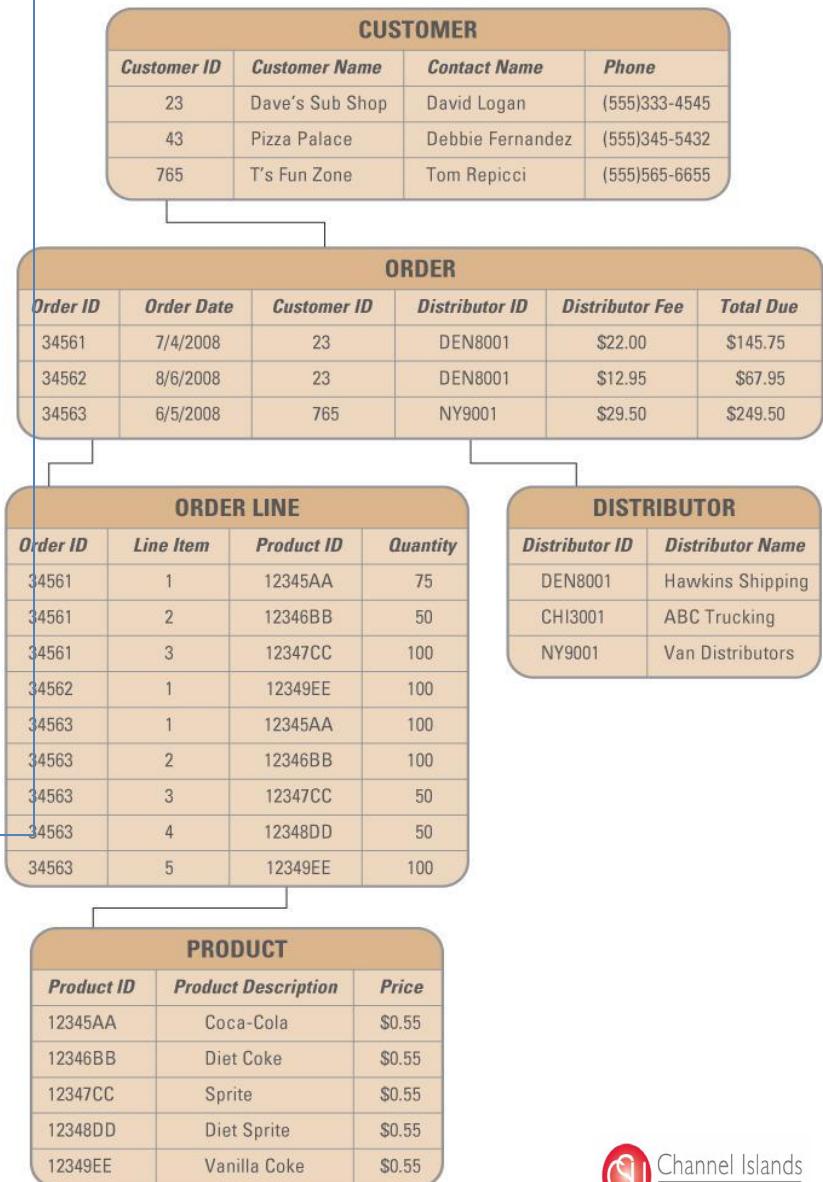


- **Primary Key:** Unique identifier for a database record.
  - Table can contain only one primary key constraint
  - Enforces the entity integrity of the table
  - Simple Key: A key contains only one column.
  - Compound Key: A key has more than one column.
- **Foreign Key:** An attribute or a set of attributes in a table which also plays the role of a primary key in another tables.
  - Acts as a cross-reference between tables because it references the primary key of another table, thereby establishing a link between them.

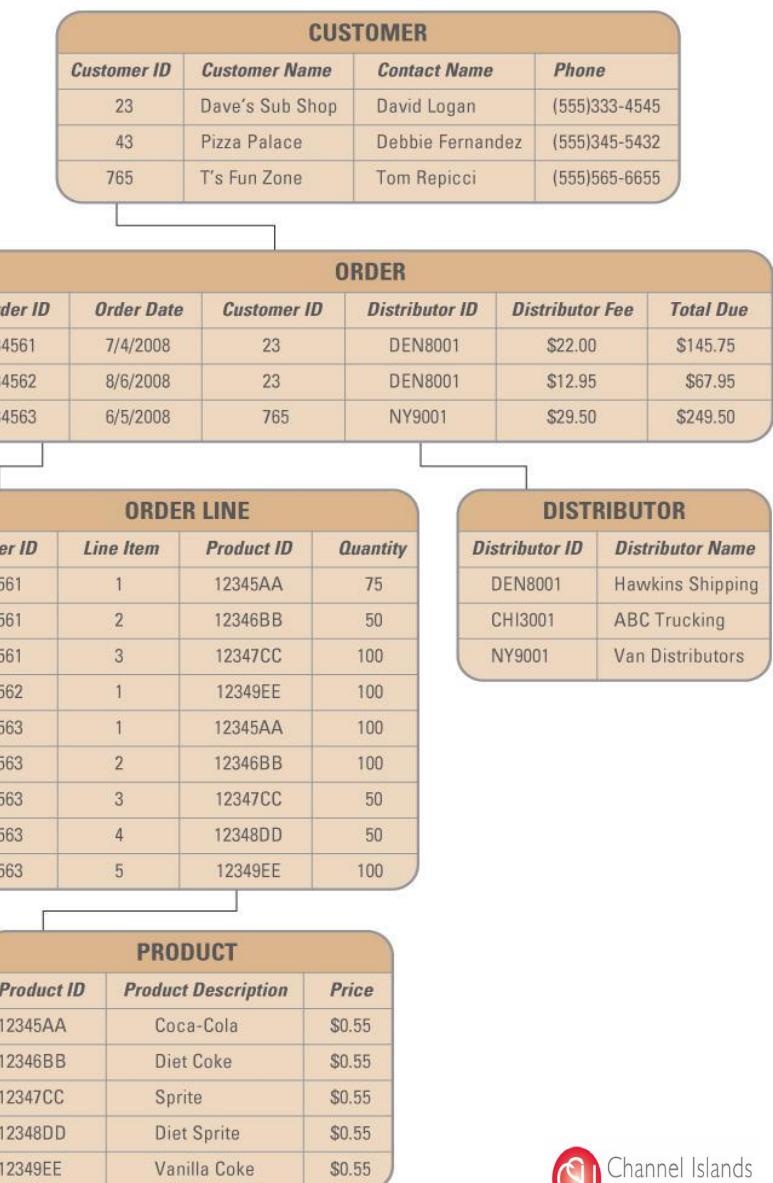
# Databases: Relationship Model

- Tables, Columns, Rows, Primary Keys, Foreign Keys and Relationships

- Each entity associated with a table.
- Attributes are columns of the table.
- Each attribute is given a data type.
- Unique identifiers are “primary keys”
- Relationships are embodied as “foreign keys”
  - An attribute whose value is the unique identifier in another table.

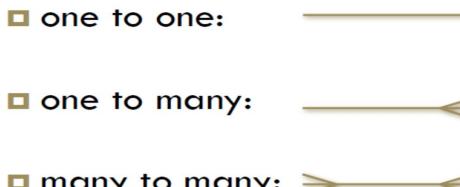


# Databases: Relationship Model



# Databases: Relationship Model



- To retrieve information from a relational database, it is necessary to associate or relate data from separate tables.
  - The three types of relationships among entities:
    - one-to-one,
    - one-to-many,
    - and many-to-many.
- 

**TABLE TB15** Rules for Expressing Relationships Among Entities and Their Corresponding Data Structures

Relationship	Example	Instructions
One-to-one	Each team has only one home stadium, and each home stadium has only one team.	Place the primary key from one table (e.g., Stadium) into the other (e.g., Team) as a foreign key.
One-to-many	Each player is on only one team, but each team has many players.	Place the primary key from the table on the “one” side of the relationship (e.g., Team) as a foreign key in the table on the “many” side of the relationship (e.g., Player).
Many-to-many	Each player participates in many games, and each game has many players.	Create a third table (e.g., Player Statistics) and place the primary keys from each of the original tables (e.g., Player and Team) together in the third as a combination primary key.

# Databases: Relationship Model

- **Index:** Data structure used to speed up data retrieval or enforce constraints on the records in a table.
  - Automatic query optimization algorithms to determine indices when searching the database.
- **Null value:** An empty value for a field.
- **View:** Logic representation of another table or combination of tables.
  - Derives its data from the tables on which it is based.
  - Permits to query subsets of data as though they are an actual table.
  - Only the view definition is stored in the database
  - Updatable view affects the underlying table.
  - Saved view is called Query in Access.

# Databases: Relationship Model

---



- **Referential Integrity:** The relationship between a referencing (child) table and a referenced (parent) table.
- Essence of a Relational Data Model:
  - Representation of Entities
  - Representation of Relationship
  - Database Integrity Rules:
    - **Primary Key Rule:** Each primary key within a table has a value that is unique.
      - No duplicates allowed.
    - **Referential Integrity Rule:** For any value in the referencing columns of a (child) table, there must exist a row in the referenced (parent) table such that the value of the referencing columns equals the value of the corresponding referenced columns.
      - Within a relational database, all values that are present on a child table should also be present on its parent table
        - » There cannot be a child without a parent
    - **Existing Rule:** Primary key cannot have a null value

# Review Questions

---

- How is information organized the relationship database?
- What are two main properties of a relation?
- How are rows that describe an entity called in the relationship database table?
- What are the other names for a column of the relationship database table?
- What do you call the unique identifier for a database record?
  - What is a foreign key?
- Explain Referential Integrity Rule in your own words.

# Databases: Data Modeling and Entity Relationship Diagram

---



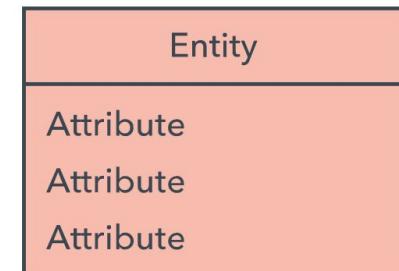
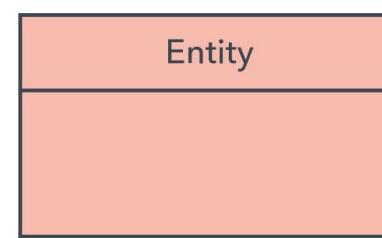
- **Data modeling** is a technique to document a software system using diagrams and symbols.
  - used to represent communication of data.
- **Entity Relationship Diagram (ERD):**
  - The highest level of abstraction for the data model
  - Graphical representation of data requirements
    - Use ERD symbols to illustrate Entities, Attributes and Relationships between Entities
    - ERD Tools
      - [Top 8 Free ER Diagram Tools in 2021](#)

# Databases: Data Modeling and Entity Relationship Diagram



## • Basic ERD Symbols

- Entity
  - Rectangle
- Attribute
  - Inside of entity's rectangle
  - or oval / circle
- Relationships between Entities
  - Lines
  - or diamonds
    - [ERD Relationship Symbols Quick Reference](#)



□ one to one:

□ one to many:

□ many to many:

# Databases: Data Modeling and Entity Relationship Diagram

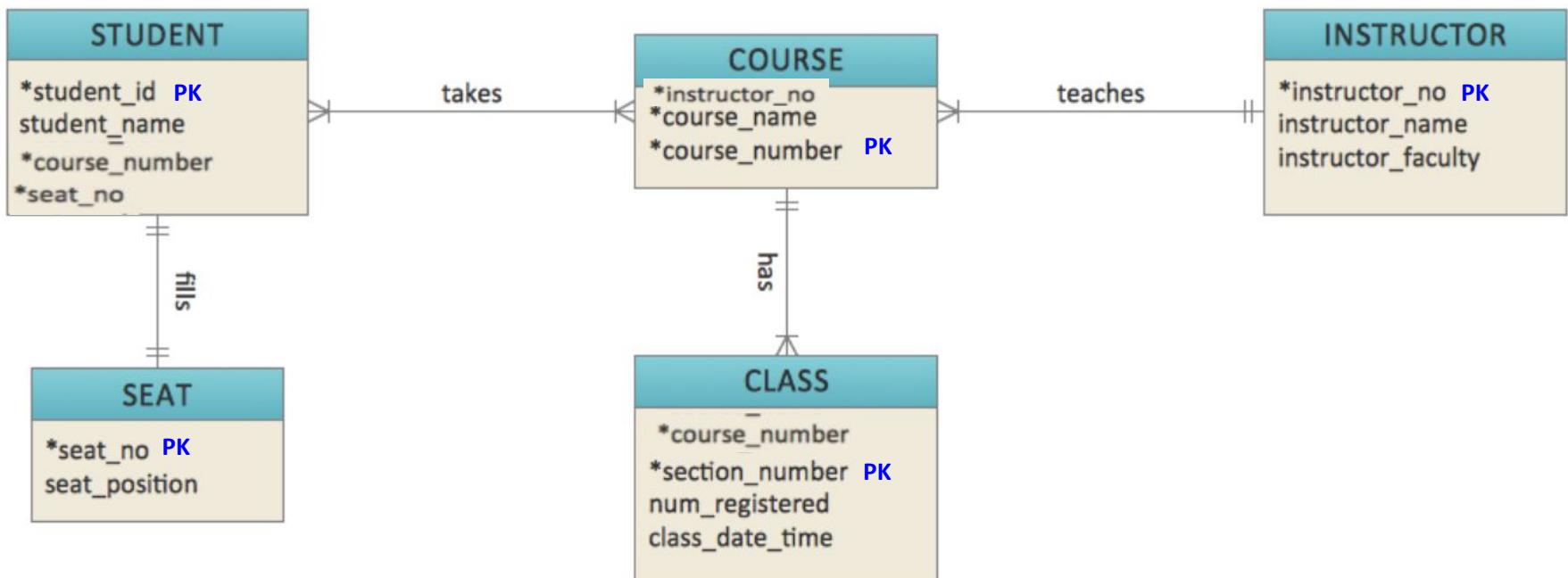
---

- **How to draw a basic ER diagram**
  - **Purpose and scope**
    - Define the purpose and scope of what you're analyzing or modeling.
  - **Entities**
    - Identify the entities that are involved, drawing them in rectangles and labeling them as nouns.
  - **Relationships**
    - Determine how the entities are all related. Draw lines between them to signify the relationships and label them as verbs directly on the connecting lines. (Can be shown as diamonds)
  - **Attributes**
    - Layer in more details by adding key attributes of entities inside of entity's rectangles. (Can be shown as an oval or circle)
  - **Primary Key**
    - Identify or create Primary Key for each Entity. Label Primary Key attribute as PK or add key symbol, or underline attribute name.
  - **Cardinality**
    - Show whether the relationship is 1-1, 1-many or many-to-many.

# Databases: Data Modeling and Entity Relationship Diagram

- **Example:**

- The database stores information on courses, classes, students, seats and instructors.
- Each student can take many classes, and each class can be taken by many students..
- One instructor can teach many courses, but one course can only be taught by one instructor.
- Each course can have many classes, but each class belongs to one course.
- Each student fills one seat and one seat is assigned to only one student.



# Databases: SQL Programming

---



- **Structured Query Language (SQL):** Standard language used to define and manipulate data in a RDBMS
  - Express actions to be performed on a relational database.
  - Standardized to allow portability across different products.
  - Can be used interactively or embedded in host languages such as C, C++, Java, PASCAL, COBOL, etc.
  - Functions on sets of records rather than individual records.
  - References to data are symbolic; they do not use any physical data structures such as pointers.
  - What is SQL?

# Databases: SQL Programming

---

- SQL is divided into three sublanguages:
  - DDL (Data Definition Language)
    - Create, alter and drop database and database objects
    - Impose integrity constraints on tables
    - Automatically update system catalog tables
    - Add comments to the tables.
  - DML (Data Manipulation Language)
    - Retrieve
    - Insert
    - Update
    - Delete rows
  - DCL (Data Control Language)
    - Grant and Revoke privileges and roles
    - Controls concurrent access to database.

# Databases: SQL Programming

---

- DDL Commands:
  - CREATE DATABASE: Creates a database.
  - CREATE TABLE: Creates a table and defines its columns and their data types.
  - CREATE SYNONYM: Creates an alternate name for a table or view.
  - CREATE VIEW: Defines a logical table, or view of data derived from columns and rows of existing tables and views.
  - CREATE INDEX: Creates an index on one or more columns of a table, for the purpose of speeding data retrieval, and enforcing uniqueness constraints.
  - ALTER: Adds, removes or renames table columns, or changes the data types of columns. Also used to create or drop UNIQUE, PRIMARY, FOREIGN KEY and CHECK constraints to enforce uniqueness, referential integrity.

# Databases: SQL Programming

---

- DDL Commands:
  - DROP DATABASE: Deletes database and all objects associated with it.
  - DROP TABLE: Deletes a table and the indexes built on that table.
  - DROP SYNONYM: Deletes an alternate name for a table or view from the system catalog.
  - DROP VIEW: Deletes a view from the system catalog.
  - DROP INDEX: Deletes a specified index. Note that an index created as the result of a UNIQUE, PRIMARY or FOREIGN KEY constraint can only be dropped by dropping the constraint through the ALTER command.
  - RENAME: Changes a table name.
  - COMMENT ON: Places a remark on a database object in the appropriate system catalog table.

# Databases: SQL Programming

---

- DML Commands:
  - SELECT: Retrieves data from one or more tables.
  - UPDATE: Modifies the data in one or more rows of a table.
  - INSERT: Inserts one or more rows into a table.
  - DELETE: Deletes one or more rows from a table.

# Databases: SQL Programming

---

- DCL Commands:
  - COMMIT: Instructs the system to make all DML commands executed by a transaction permanent.
  - ROLLBACK: Instructs the system to reverse all DML commands executed by a transaction.
  - GRANT: Assigns database access privileges to database users.
  - REVOKE: Cancels database access privileges from database users.
  - LOCK: Permits users to explicitly acquire table locks.
  - UNLOCK: Used to unlock a table prior to the commit point.

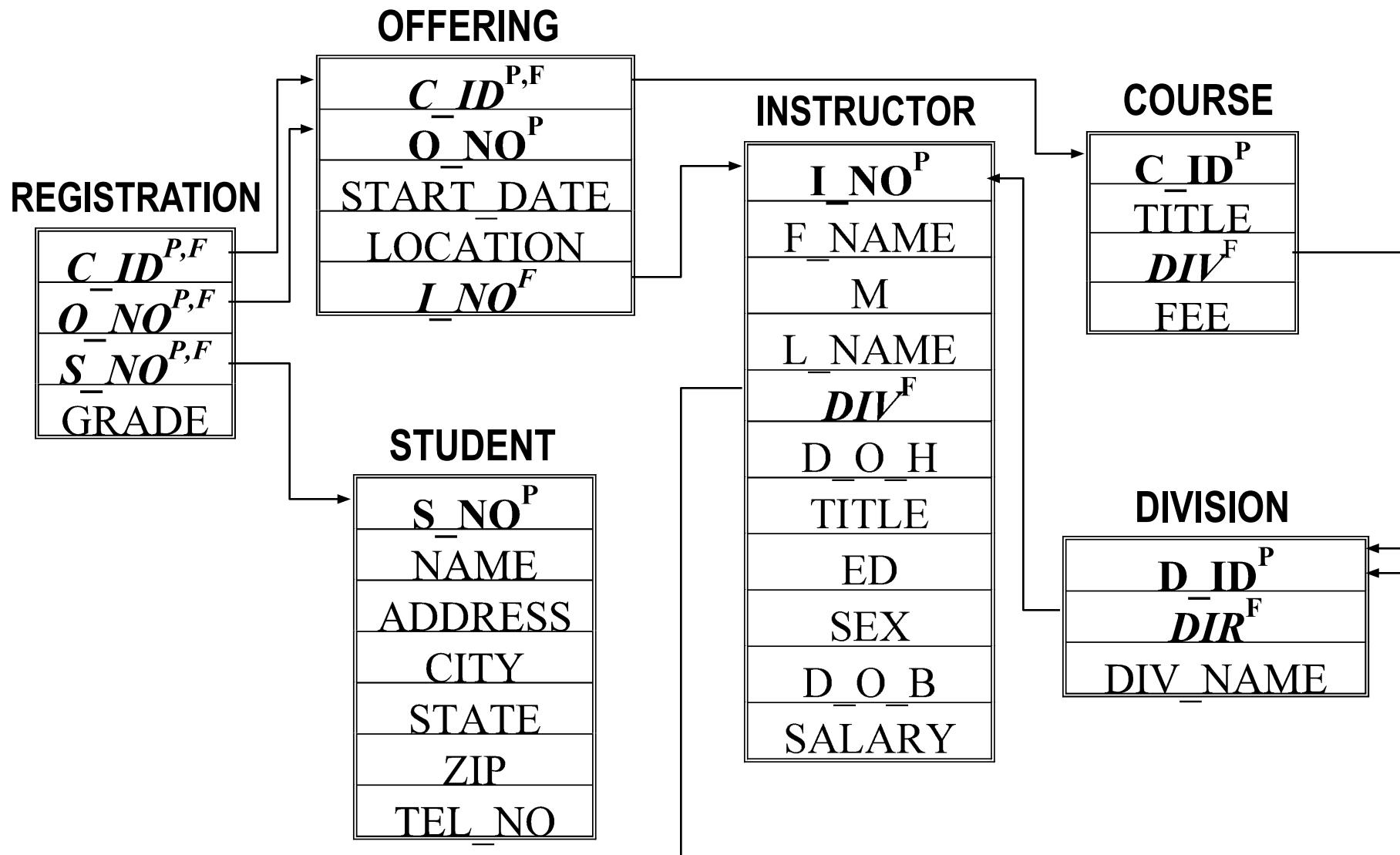
# Databases: SQL Programming

---

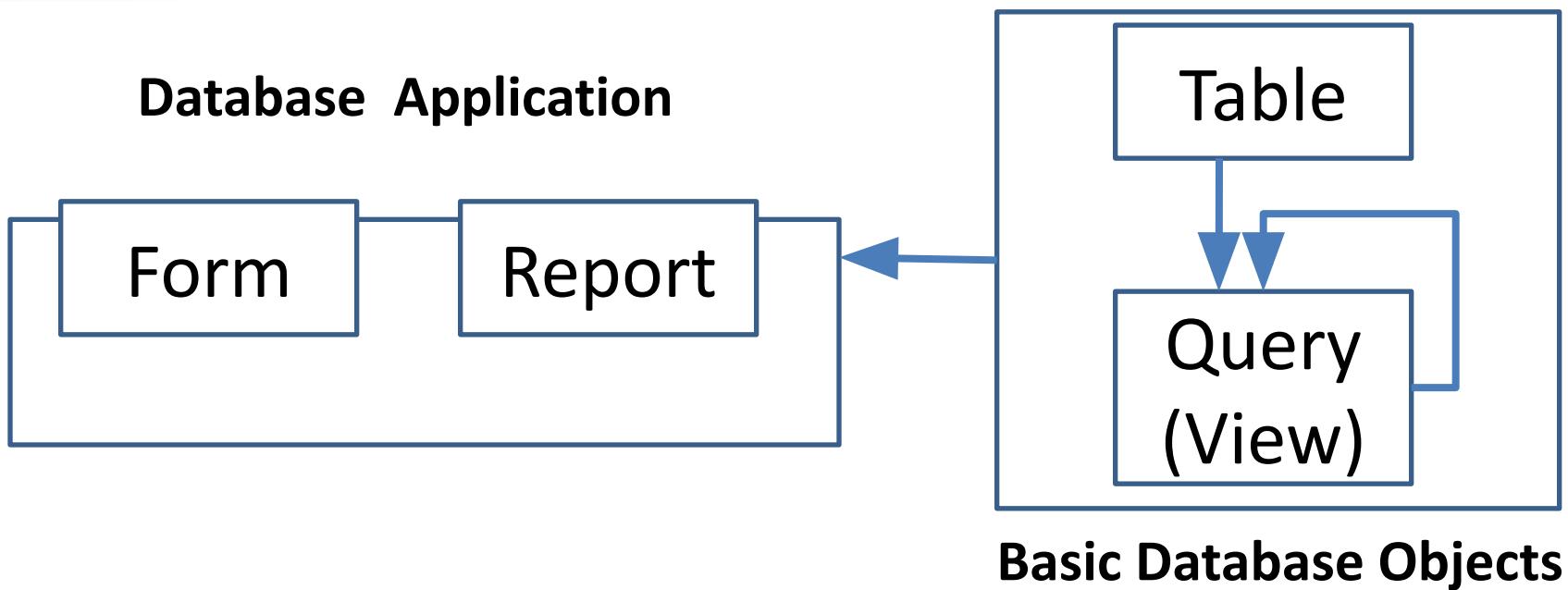
- MySQL
  - Open-source
  - Great for small to mid-sized organizations.
  - Fast, efficient, cheap
  - Doesn't support full SQL but a good portion of it.



# Databases: Sample Structure



# Databases: Objects and Applications



- You can create a query against a table or a query or both
- You can create a form or report against a table or a query or both

# Databases: MS Access Objects

- MS Access Objects

- Tables

- used to hold all the information or data

- Forms

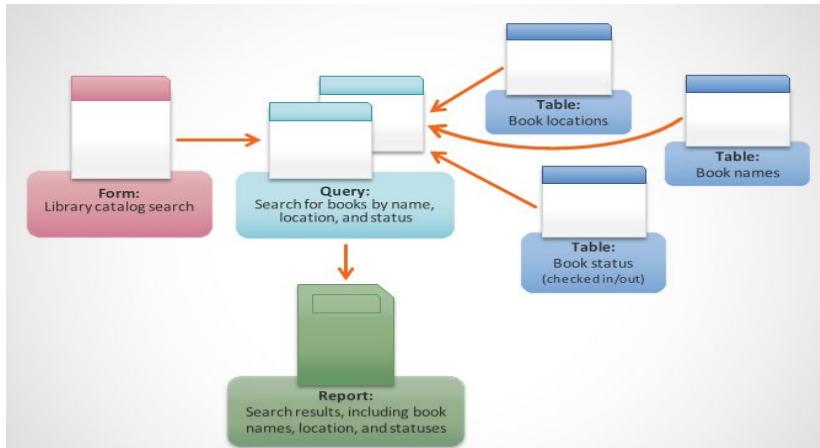
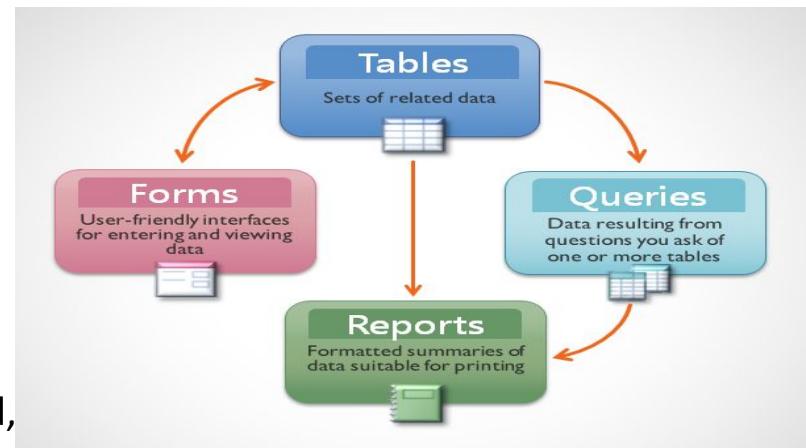
- used to capture data to be added, modified, or deleted from a database by the users.

- Queries

- used to retrieve data from a database

- Reports

- present the information in a database

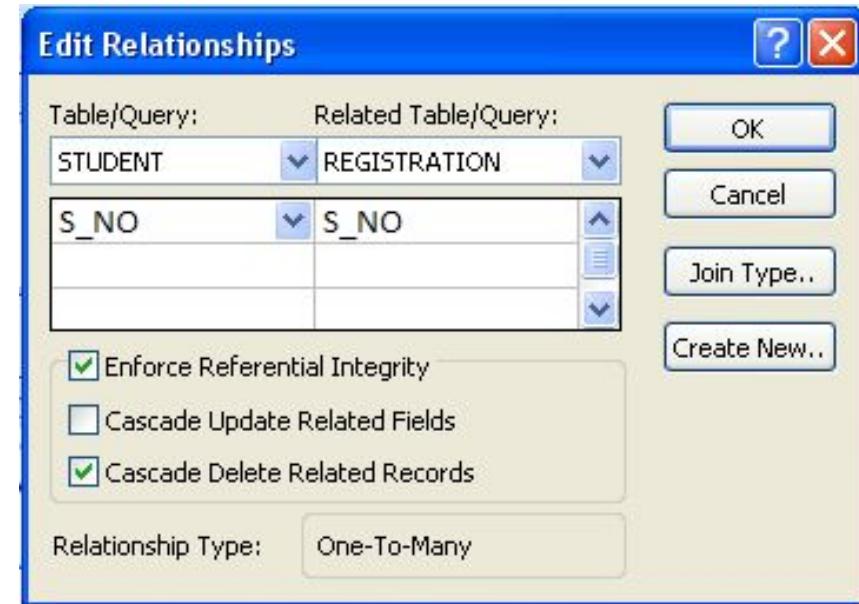


# Databases: Relationships

- Define Relationships in Access

- Referential Integrity

- You can't enter a value in the Foreign Key field of one (Child) table if there is not a matching value in the Primary Key of the related (Parent)table.
    - You can't delete a record from the primary (Parent) table if a matching record exists in the related (Child) table.
    - You can't change the value in the Primary Key of the Parent table if there are related records in the Child table



- Join Type

- One to One
    - One to Many

# Databases: Create Table

- Create Table in MS Access
  - Switch to Design View
  - Identify fields

Field Name	Data Type
D_ID	Text
DIR	Number
DIV_NAME	Text

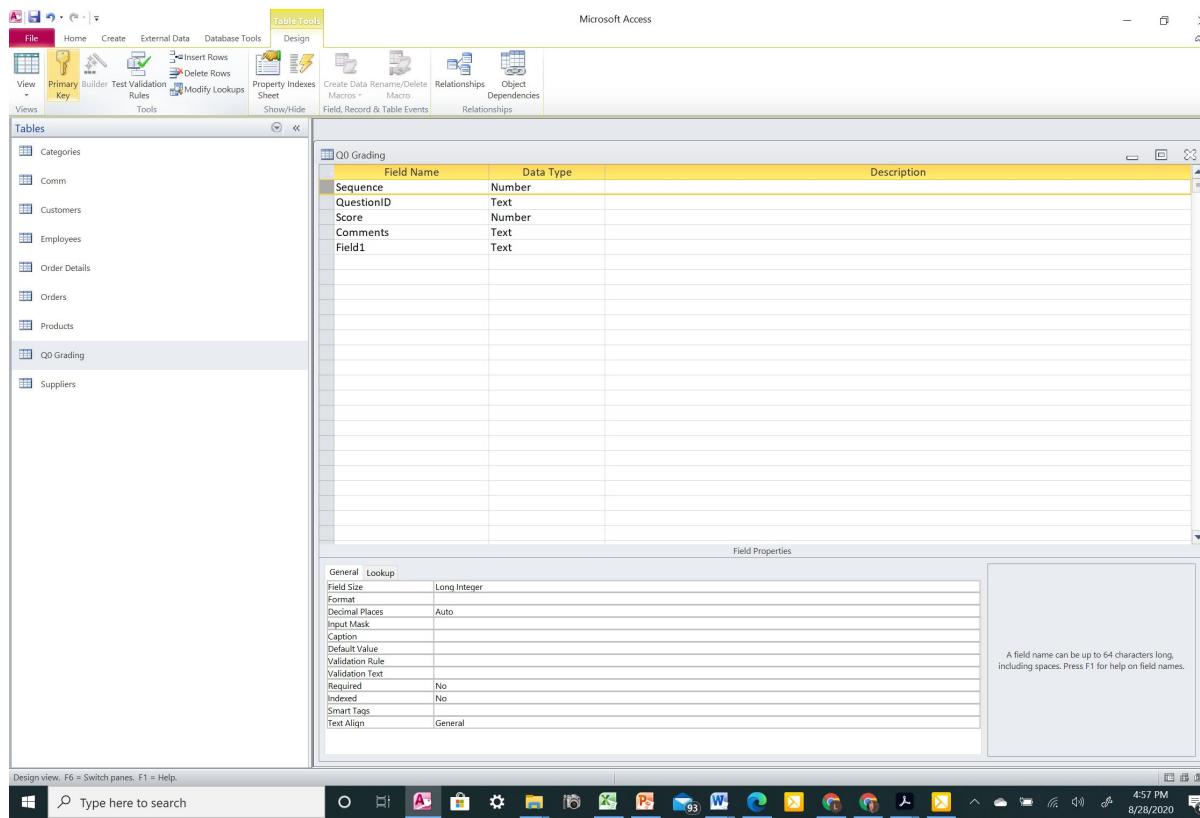
**General** **Lookup**

Field Size	Long Integer
Format	
Decimal Places	
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	

Field Size: Long Integer  
Format: Byte  
Decimal Places: Integer  
Input Mask: Long Integer  
Caption: Single  
Default Value: Double  
Validation Rule: Replication ID  
Validation Text: Decimal

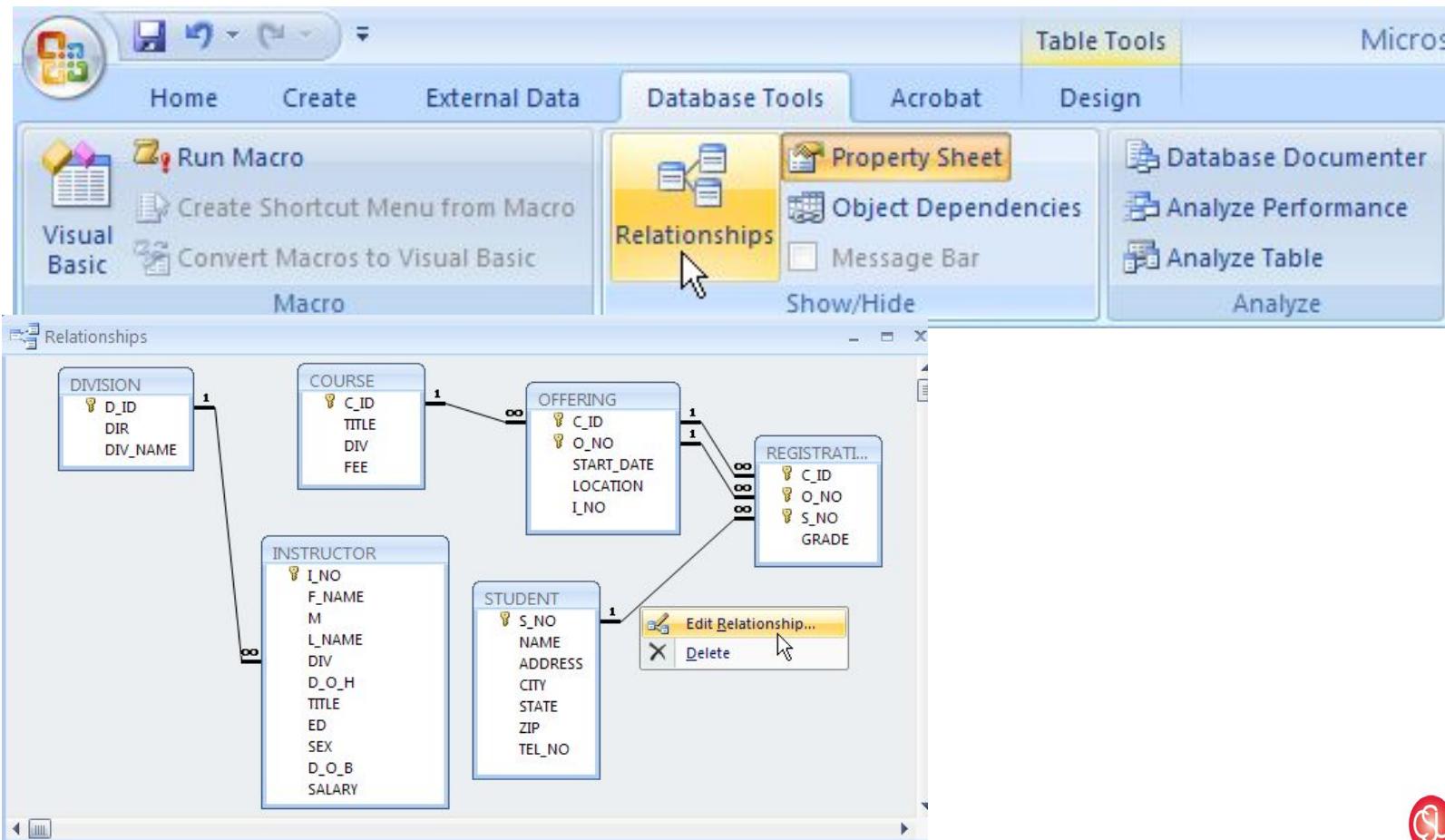
# Databases: Create Table

- Create Table in MS Access
  - Assign Primary Key
    - Highlight the field
    - Click on Primary Key in toolbar menu



# Databases: Relationships

- Define Relationships in MS Access
  - To create relationship drag the primary key of a table into foreign key of another table



# Databases: Normalization

---

- First Normal Form

- No repeating value.

*Un-Normalized Relation:*

*Normalized Relation:*

STUDENT	COURSE
Tom	C102
Sam	{C102, C103, C105}
Ralph	{C102, C105}

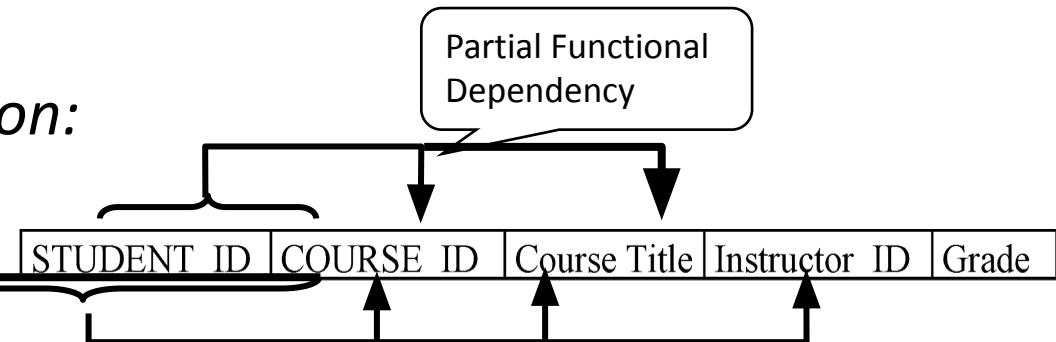
STUDENT	COURSE
Tom	C102
Sam	C102
Sam	C103
Sam	C105
Ralph	C102
Ralph	C105

# Databases: Normalization

- **Second Normal Form**

- No any partial functional dependencies.
- Non-identifying attributes are dependent on the entity's unique identifier.

*Un-Normalized Relation:*



*Normalized Relation:*

*Registration*

STUDENT ID	Course ID	Grade

*Course*

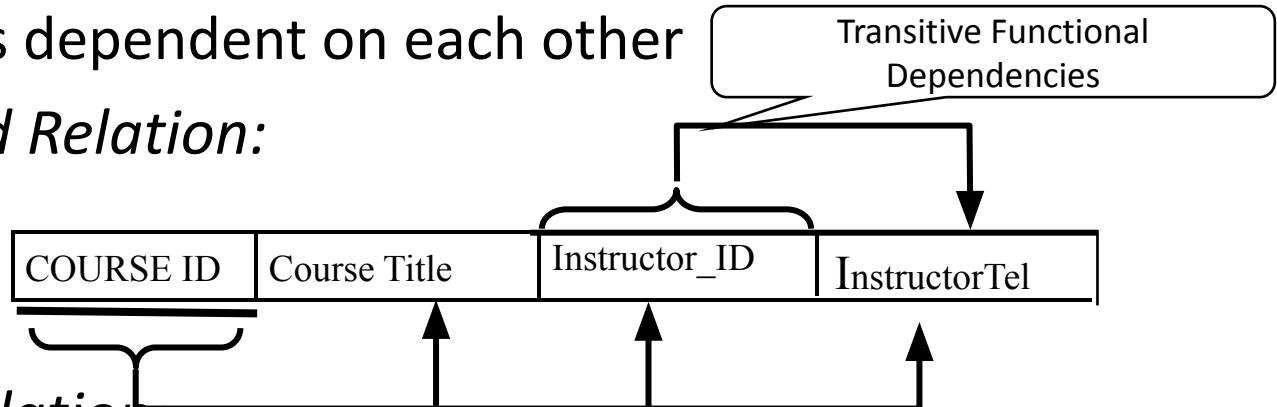
COURSE ID	Cours Title	Instructor ID

# Databases: Normalization

- **Third Normal Form**

- No transitive functional dependencies
- No attributes dependent on each other

*Un-Normalized Relation:*



*Normalized Relation:*

*Course*

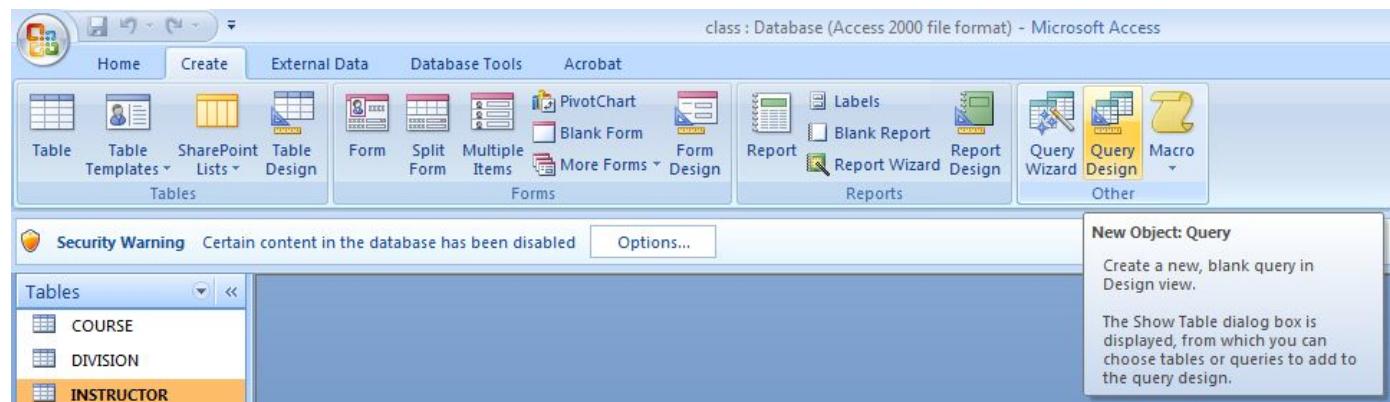
COURSE ID	Cours Title	Instructor ID

*Instructor*

INSTRUCTOR ID	Instructor Tel

# Databases: Query

- Create Query in MS Access
  - To select data from Single or Multiple Tables or Queries or Both
  - Saved View is called Query in MS Access database
  - QUERY in MS Access can be SELECT, INSERT, UPDATE, APPEND or DELETE.



**Highlight a table or a query, click Add; or  
Double click on a table or a query.  
You may select more than one table  
when  
performing SQL JOIN**



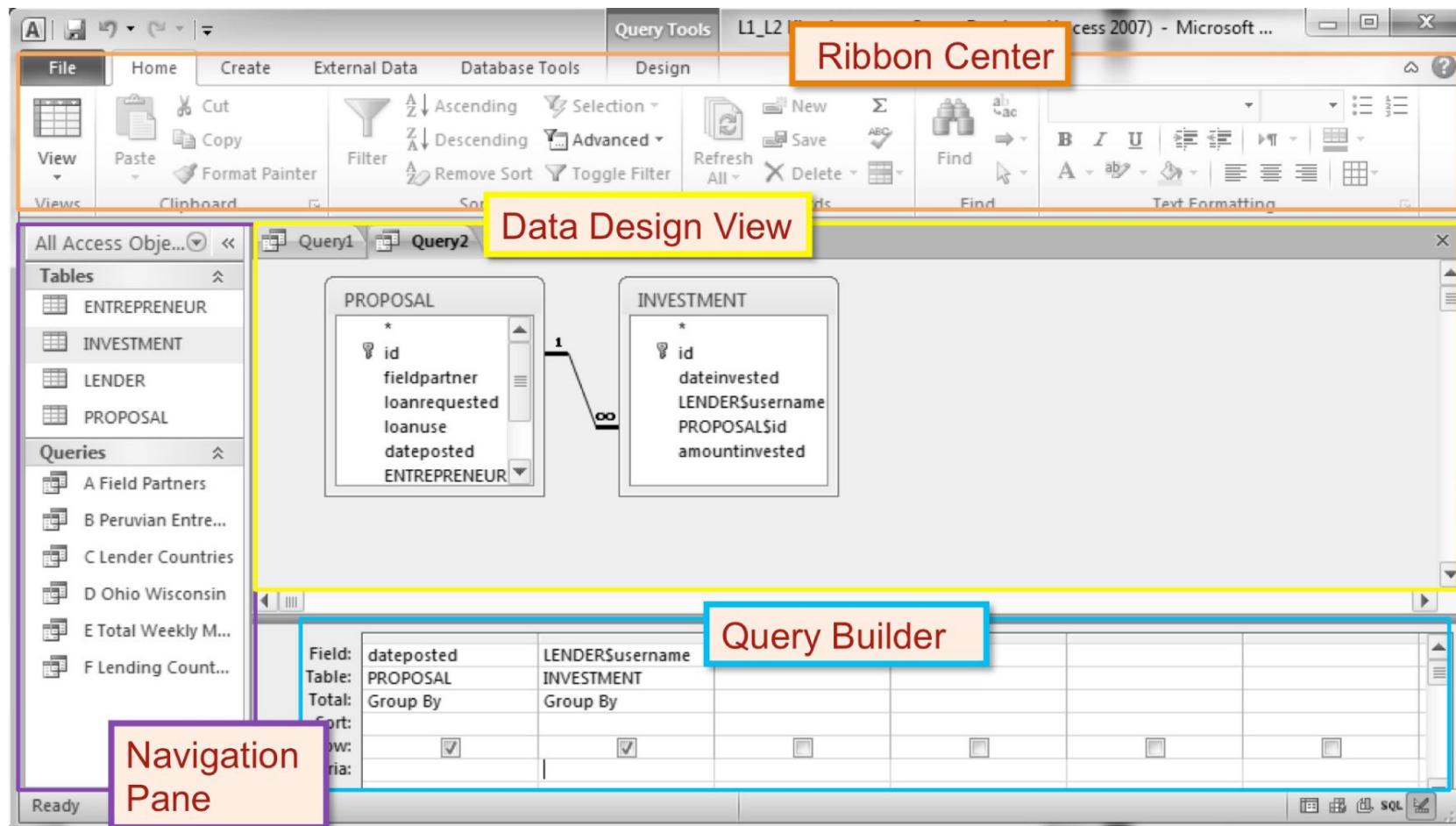
# Databases: Query

- Create Query in MS Access
  - Switch to Design view
  - Add table/s
  - Specify criteria

The screenshot shows the Microsoft Access Design View window titled "Query2". In the top pane, there is a table named "COURSE" with fields: C\_ID, TITLE, DIV, and FEE. The "C\_ID" field is marked with an asterisk (\*) indicating it is the primary key. In the bottom pane, the query definition is displayed:

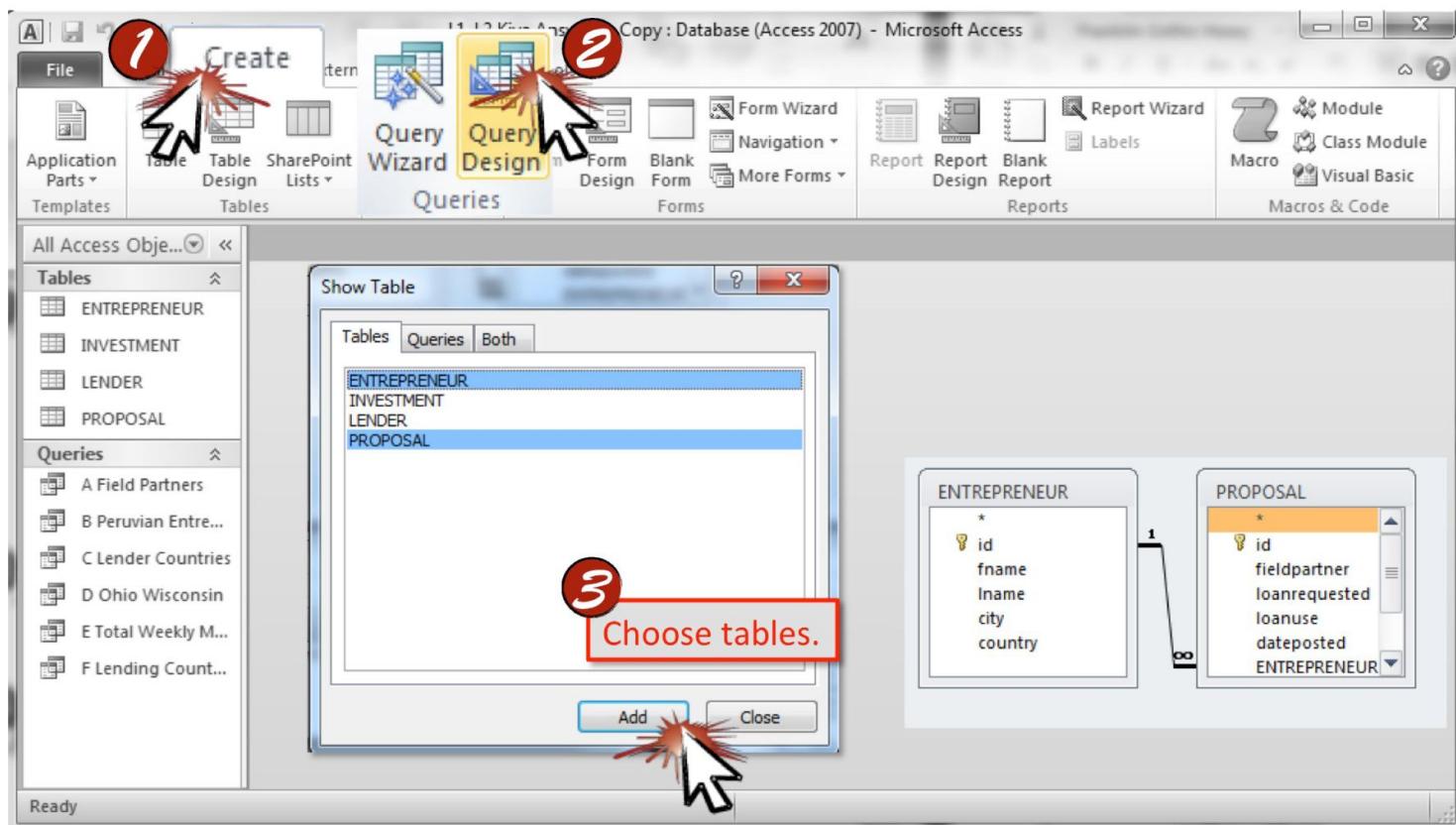
Field:	C_ID	TITLE	FEE		
Table:	COURSE	COURSE	COURSE		
Sort:			Descending		
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Criteria:			>250 And <=350		
or:					

# Databases: Query



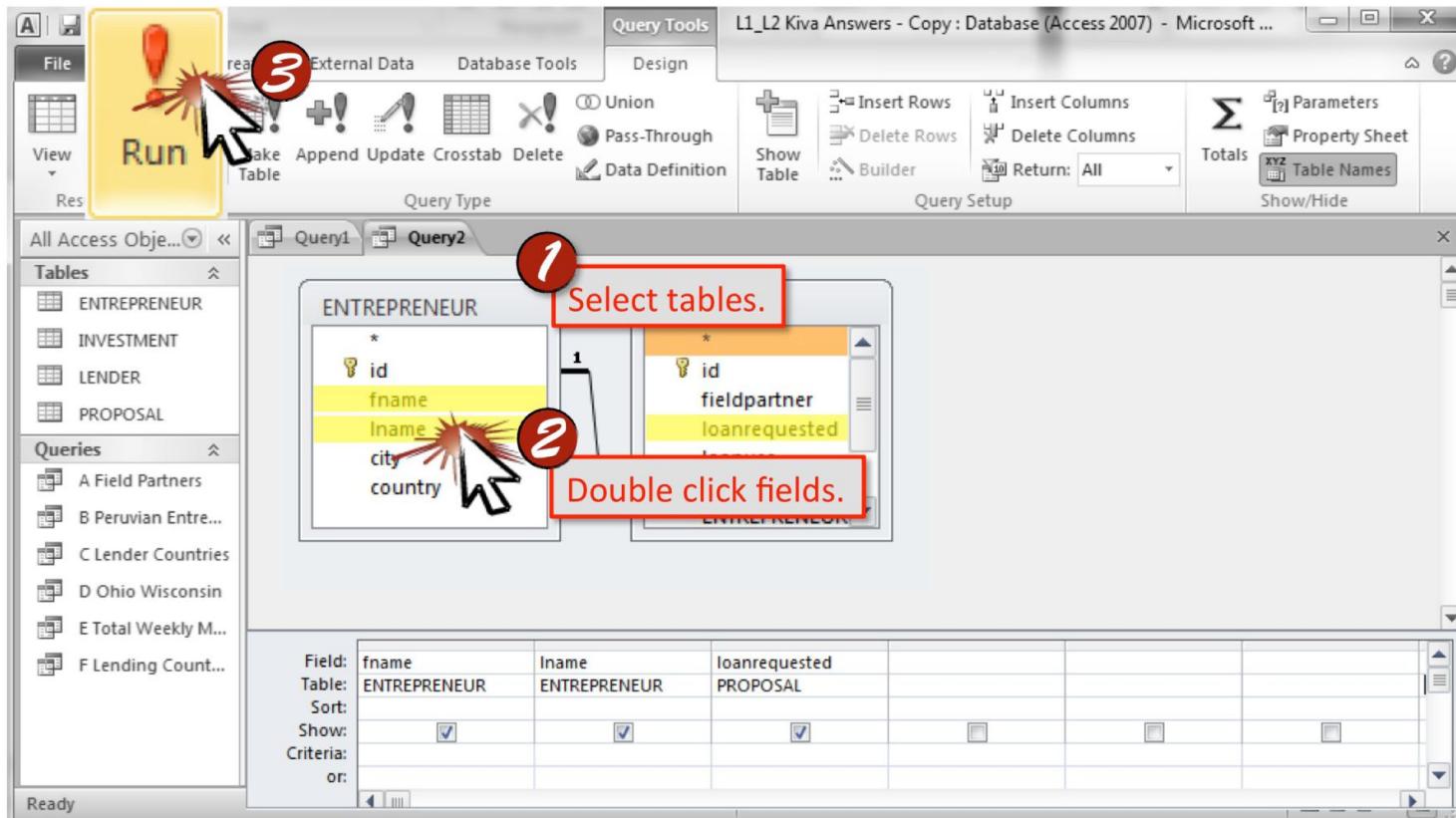
**Overview Map of Interface**—Reference this map to navigate the Microsoft Access interface.

# Databases: Query



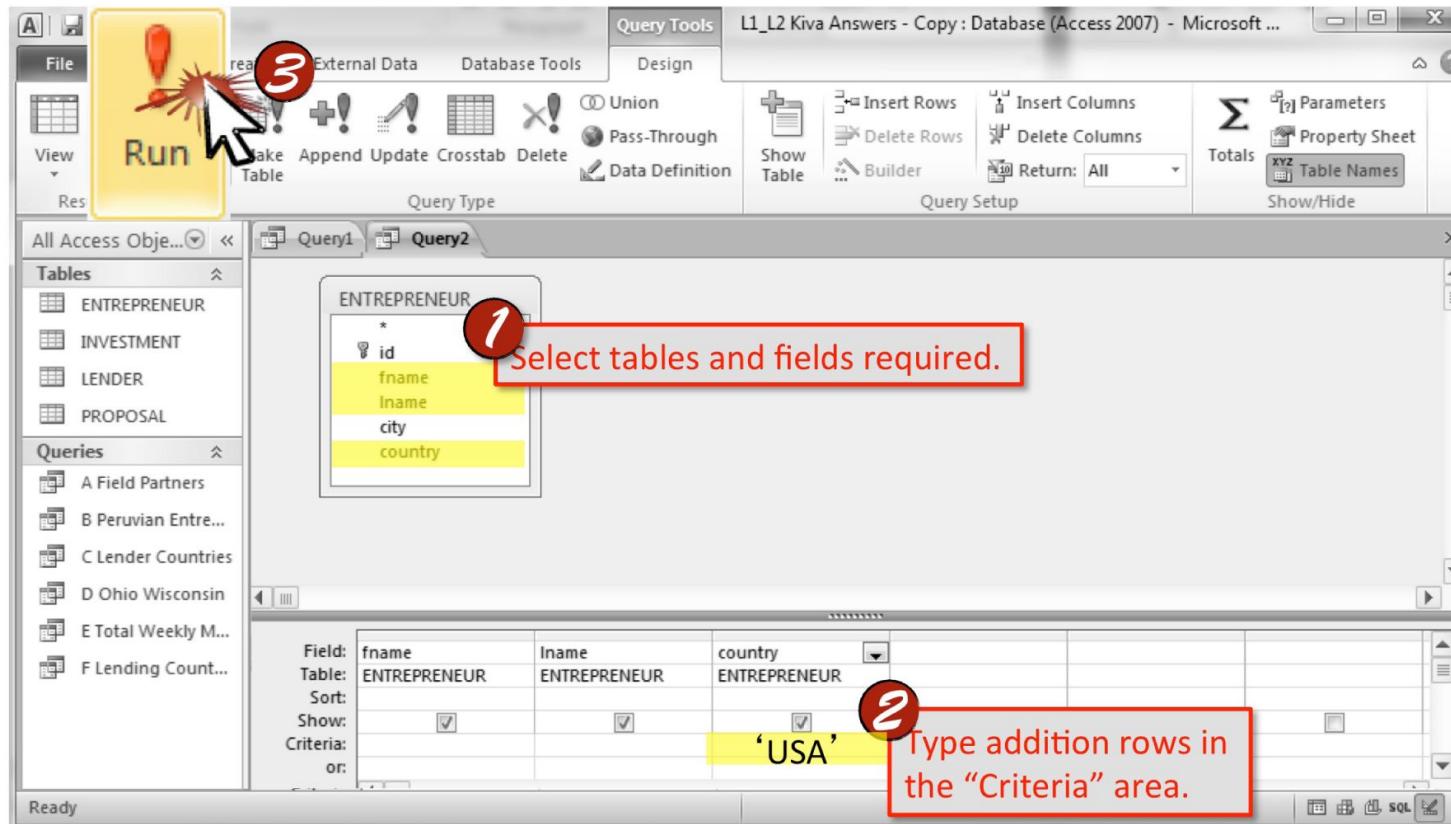
**Select Tables**—Choose the tables that contain data you will be using in your query. Some queries will require only one table while others will require multiple tables from your database.

# Databases: Query



**Select Fields**—Choose the individual fields that you will require when constructing your query. Consider what you are asking and which pieces of data are necessary to achieve the answer.

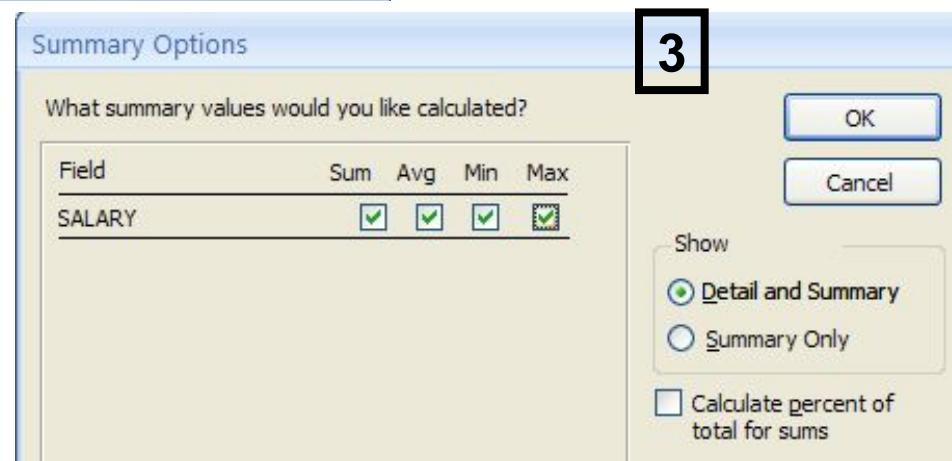
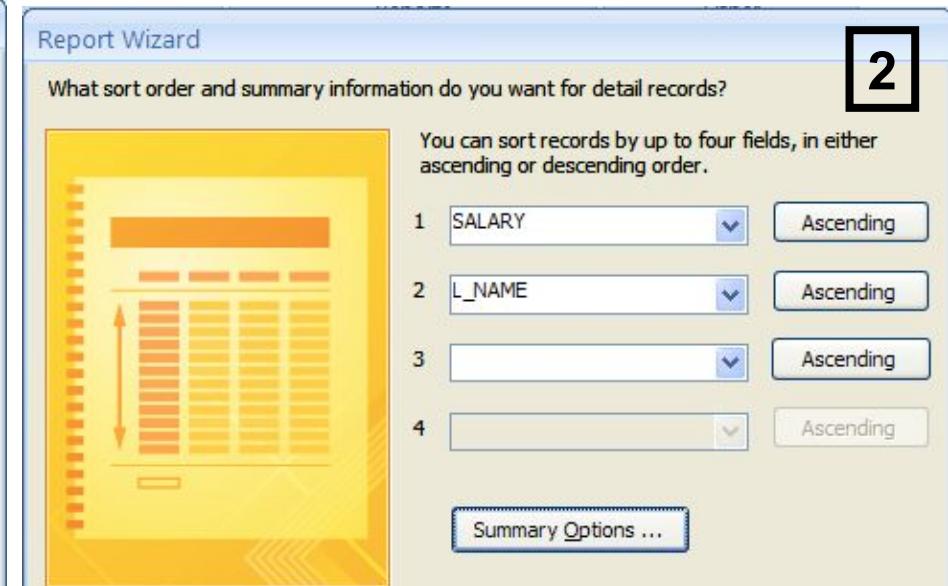
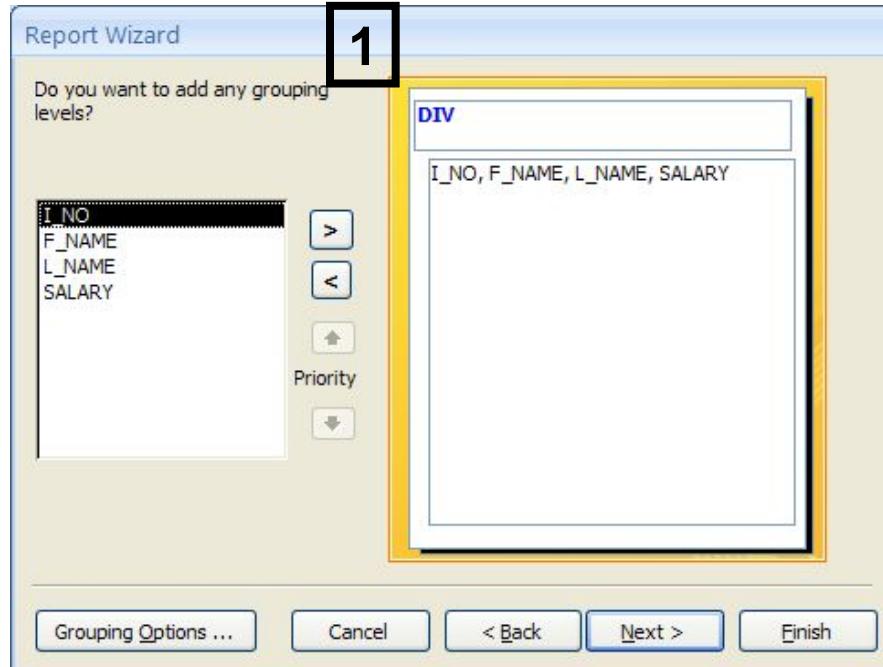
# Databases: Query



**Select Rows**—Add criteria to your query to narrow your search. By including additional criteria, you will be able to obtain more powerful information from your data.

# Databases: Reports

- Access Report Wizard



# Databases: Forms

- MS Access Forms

This screenshot shows two forms in Microsoft Access:

- DIVISION Form:** The top form has fields for D\_ID, DIR, and DIV\_NAME. It also contains a subform titled "Division Instructor Master Detail" which displays data for multiple instructors.
- INSTRUCTOR Form:** The bottom form has fields for I\_NO, F\_NAME, and L\_NAME.

This screenshot shows the Microsoft Access **Design View** for the DIVISION form. A callout arrow points from the "Form View" icon in the ribbon to the main window. Another callout arrow points from the "Design View" icon in the ribbon to the subform area of the DIVISION form.

The DIVISION form includes the following fields and subform:

- Master Fields:** D\_ID, DIR, and DIV\_NAME.
- Subform:** Division Instructor Master Detail, which contains the following fields:

I_NO	F_NAME	L_NAME	SALARY	DIV
1	PATRICK	LEE	72000	D020
19	ALBERT	STONE	60000	D020
21	BRUCE	WANG	48000	D020

Sources:

Joseph Valacich, Christoph Schneider, *Information Systems Today: Managing in the Digital World*, 8th Edition  
John Gallagher, Information Systems: A Manager's Guide to Harnessing Technology, v. 7.0  
Minder Chen, Ph.D., Management Information Systems Lectures