NATIONAL ENGINEERING CENTER
University of the Philippines
Diliman, Quezon City

Introduction to Databases: A Review

Eugene Rex L. Jalao, Ph.D.

Associate Professor

Department Industrial Engineering and Operations Research

University of the Philippines Diliman

Module 2 of the Business Intelligence and Analytics Track of UP NEC and the UP Center of Business Intelligence

Outline for This Training

1. Introduction to Data Warehousing
2. DW Lifecycle and Project Management
   - Case Study on DW PM
3. Dimensional Modeling
4. Designing Fact Tables
5. Designing Dimension Tables
   - Case Study on Dimension Modeling
6. Extraction Transformation and Loading
   - Case Study on ETL Planning
7. Transformation and Loading Methodologies
   - Case Study on ETL

eljalao@up.edu.ph

Outline for this Session

- What is a Database?
- How are databases structured?
- How do we manipulate data in a database?

eljalao@up.edu.ph

What is a Database?

Definition 1: Files

- A **file** is a collection of related information
- A system of files and collection of application programs manipulating them is a **file-based system**

eljalao@up.edu.ph

What is a Database?

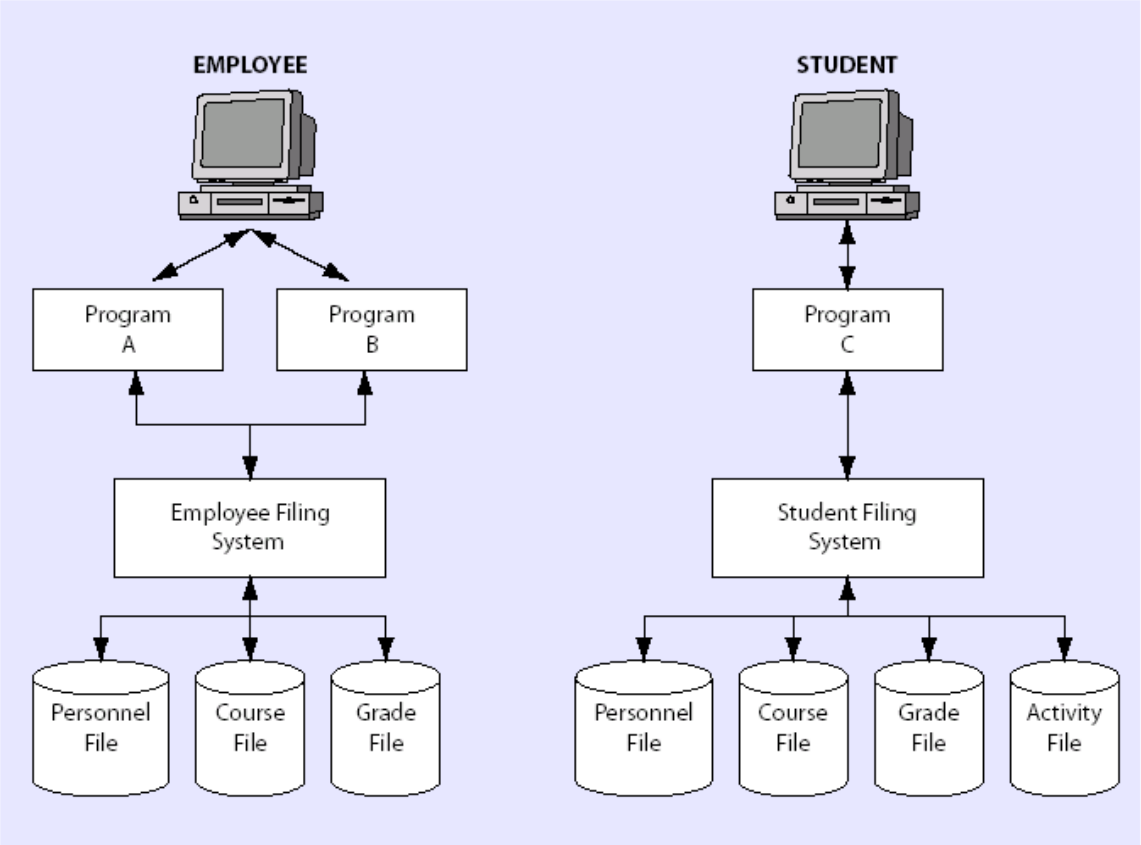


Figure 1: A University's File-Based System

What is a Database?

- Limitations of the File-Based Approach
  - Efforts for **query** answering:
    - What is the average grade for Dr. Rex's students?
    - List the activities for all students enrolled in EZZ 4162.
    - Which personnel are students as well as staff?
  - Other **limitations**:
    - Duplication of data
    - Data dependency
    - Slow development, high maintenance and fixed queries

What is a Database?

- Limitations of file-based approach tell us that:
  - Parameters defining data should be **separately stored**
  - There should be a way to control and manipulate data in **isolation** of the application program

- A database is a **computer solution** for fast, efficient, accurate, and secure data access

eljalao@up.edu.ph

What is a Database?

Definition 2: Files

- A database is a **computer solution** for fast, efficient, accurate, and secure data access

eljalao@up.edu.ph

What is a Database?

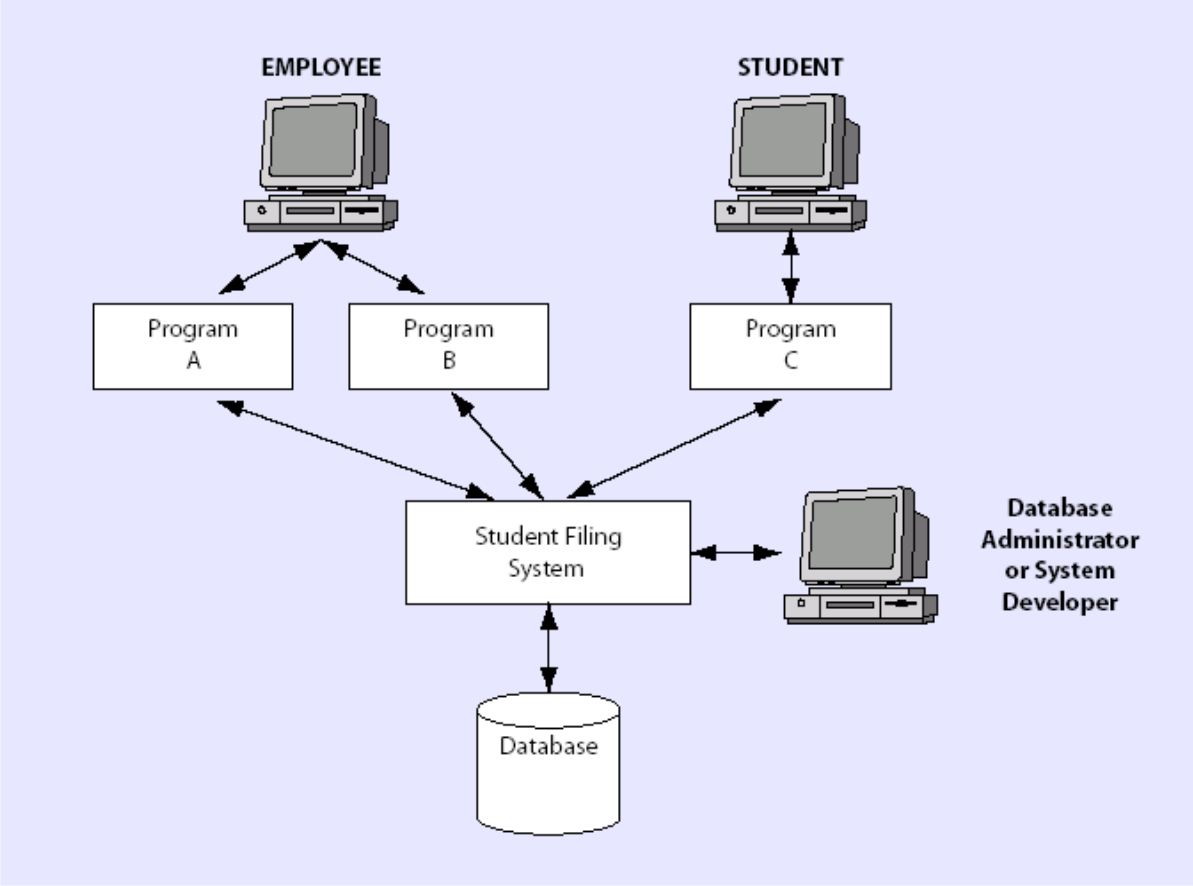


Figure 2: A University's Database System

What is a Database?

- There are **four** components in any database system:
  - Users
  - Database application
  - DBMS
  - Database

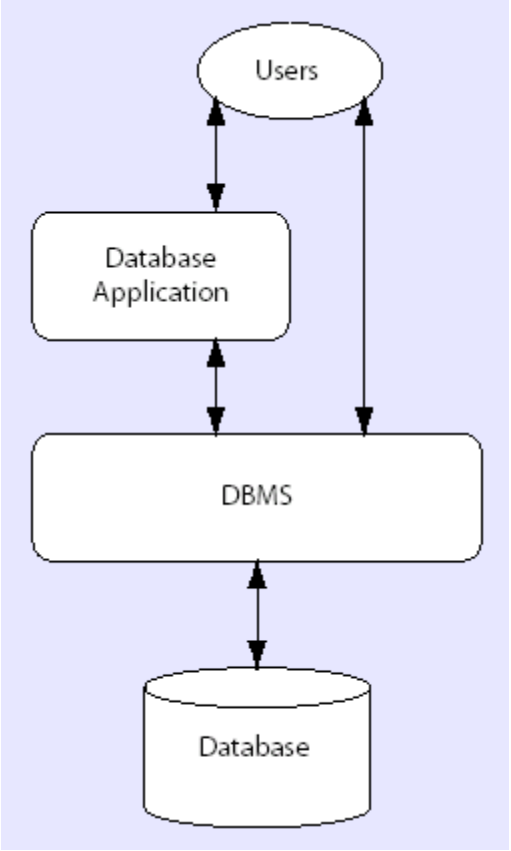


Figure 3: Components of a Database System

What is a Database?

- A database system can be defined as the **combination** of a database, a DBMS, and application programs
  - Because of the advantages offered by a DBMS, businesses and organizations prefer the database approach to the file-based approach
- A database management system (DBMS) is a piece of software that allows a user to define, create, and manage access to a database
  - Decouples application programs from data
  - The database stores all its data in one location, thereby limiting data duplication
  - Access, Oracle, IBM's DB2, and SQL Server

What is a Database?

- Database applications:
  - Computer programs that allow users to manipulate the data in a DBMS through a user-friendly interface
  - Can be divided into **four broad** categories:
    - Personal: Restricted to a single user
    - Departmental: Referenced by hundreds of users over a shared system or network
    - Enterprise: Extensions of departmental applications involving thousands of users
    - Internet: Largest form of information sharing where billions of users are involved

eljalao@up.edu.ph

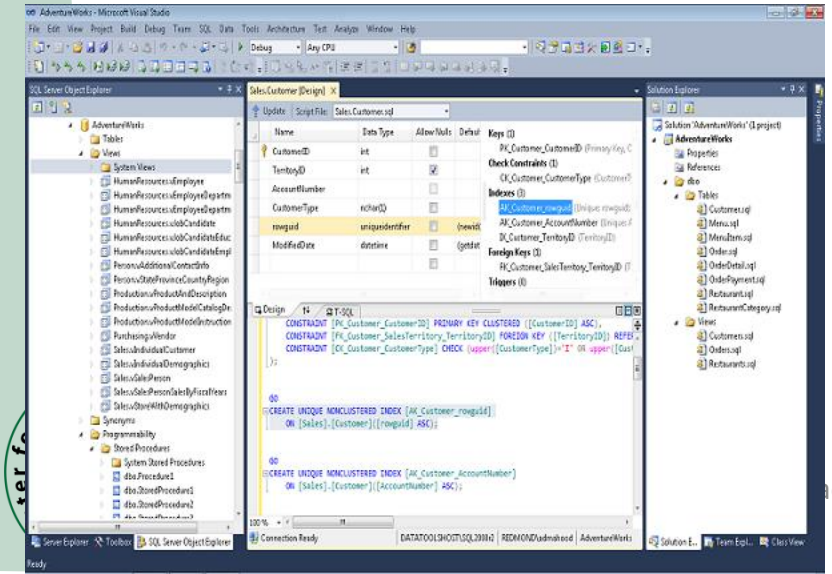
What is a Database?

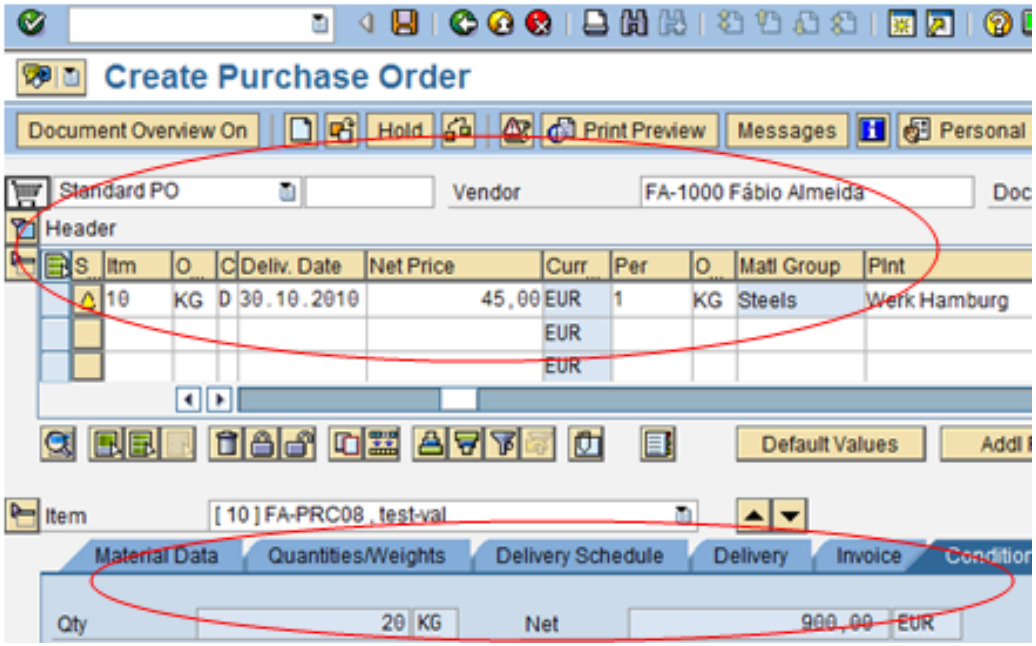
- Database administrator, system developer, and end user:
  - A **database administrator** (DBA) is a person responsible for all the data resources of an organization
    - Uses tools that come with a DBMS to improve the productivity and performance of database planning and design
  - **System developers** are a group of people responsible for the creation of new application programs that cater to the user requirements
    - Use their own tools to write programs that communicate with the DBMS

What is a Database?

- Database administrator, system developer, and end user:
  - **End-users** in an organization can add, update, and delete data in a database through application programs or directly through a DBMS
    - Use the application program to accomplish their day-to-day tasks

What is a Database?





eljalao@up.edu.ph

What is a Database?

- Functions of a DBMS
  - A DBMS is primarily responsible for providing a **logical view** of underlying data
    - Allows its user to store, retrieve, and update data
    - Provides a clear and logical view of the process that manipulates the data
  - Other **functionalities**:
    - Data independence
      - Maintain segregation between the program and the data
    - Concurrency control
    - Recovery services
    - Utility services
      - Perform initialization and maintenance operations on a database

eljalao@up.edu.ph

What is a Database?

- **Advantages:**
  - Segregation of the application program and the data
  - Minimal data duplication
  - Ability to retrieve data easily
  - Reduced development time and maintenance needs
- **Disadvantages:**
  - Complexity
  - Size
  - Cost

Outline for this Session

- What is a Database?
- **How are databases structured?**
- SQL

eljalao@up.edu.ph

How are Databases Structured?

- Designed to **eliminate redundancies**. Other than keys, each attribute may appear in only one table.
- Design objective: a **Third Normal Form (3NF)** model.
- Modeling business processes results in **numerous data entities/tables** and a spaghetti-like interweaving of relationships among them.
  - Some ERP systems have tens of thousands of tables.
  - Even a small model can be challenging.

How are Databases Structured?

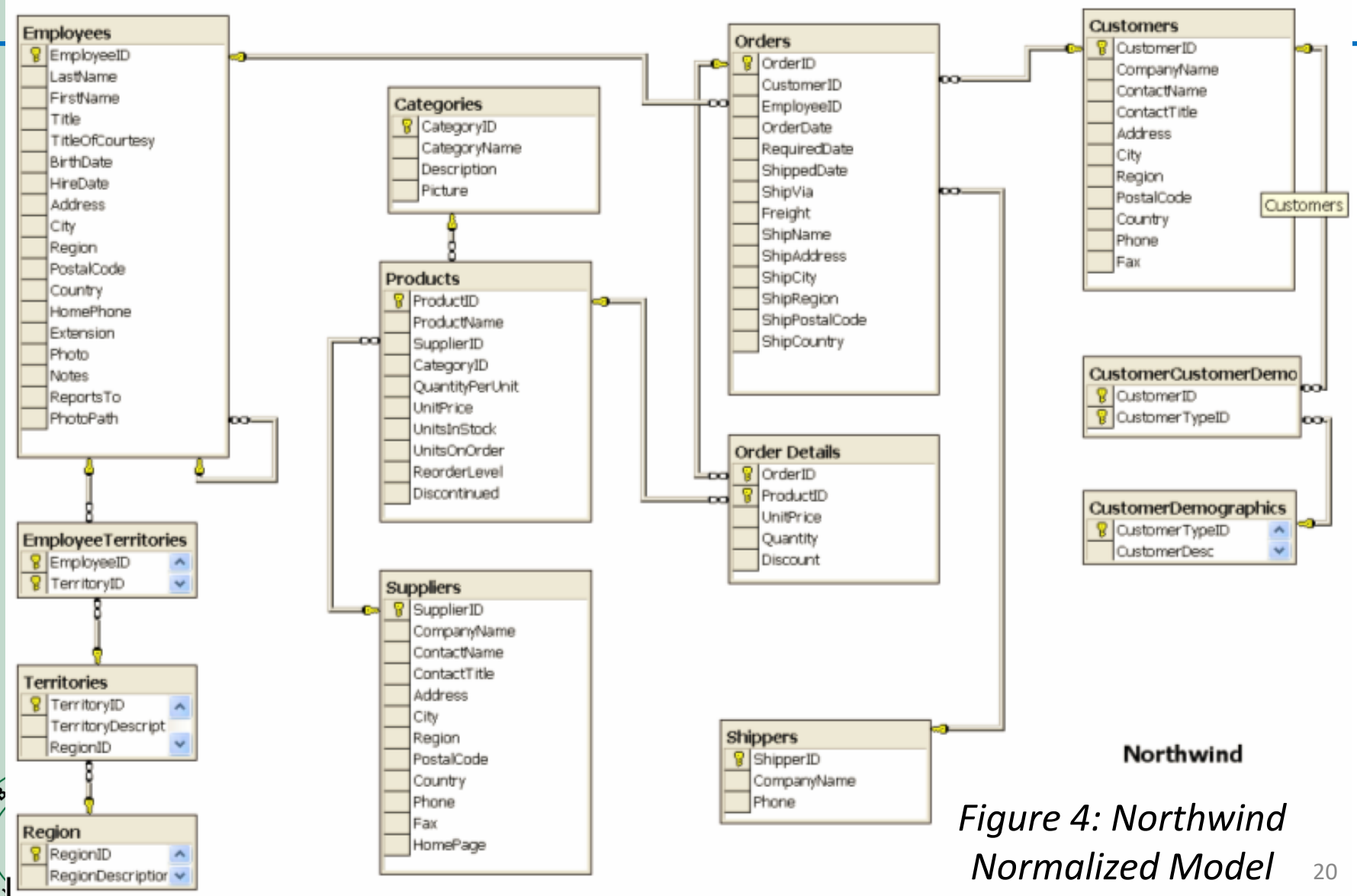


Figure 4: Northwind Normalized Model

How are Databases Structured?

Definition 3: Normalization

- Normalization is the systematic process of simplifying and generalizing the structure of data stores to better accommodate **future changes.**
- Design objective: a **Third Normal Form (3NF)** model.

How are Databases Structured?

- Normal Forms
  - Zeroth Normal Form
  - 1st Normal Form
  - 2nd Normal Form
  - 3rd Normal Form

eljalao@up.edu.ph

How are Databases Structured?

Definition 4: Keys

- A key is a single data element or a combination of data elements which **uniquely identifies each record** in a table
- A key consisting of more than one data element is called a **Concatenated Key**

How are Databases Structured?

Definition 5: Relationships

- Tables are related to each other through **relationships** known as **joins**.
  - Relationships are diagrammed as lines with symbols at one or both ends to indicate the nature of the relationship.
- Most common relationship: **one-to-many**,
  - each row in the one table can have zero-to-many rows related to it in the many table.
    - One Purchase Order-> Multiple Purchase Items

How are Databases Structured?

Definition 6: Entity-Relationship Model

- An **entity-relationship model** describes data in terms of the following:
  - Entities
  - Relationship between entities
  - Attributes of entities
- We graphically display an E-R model using an **entity-relationship diagram** (E-R diagram)

How are Databases Structured?

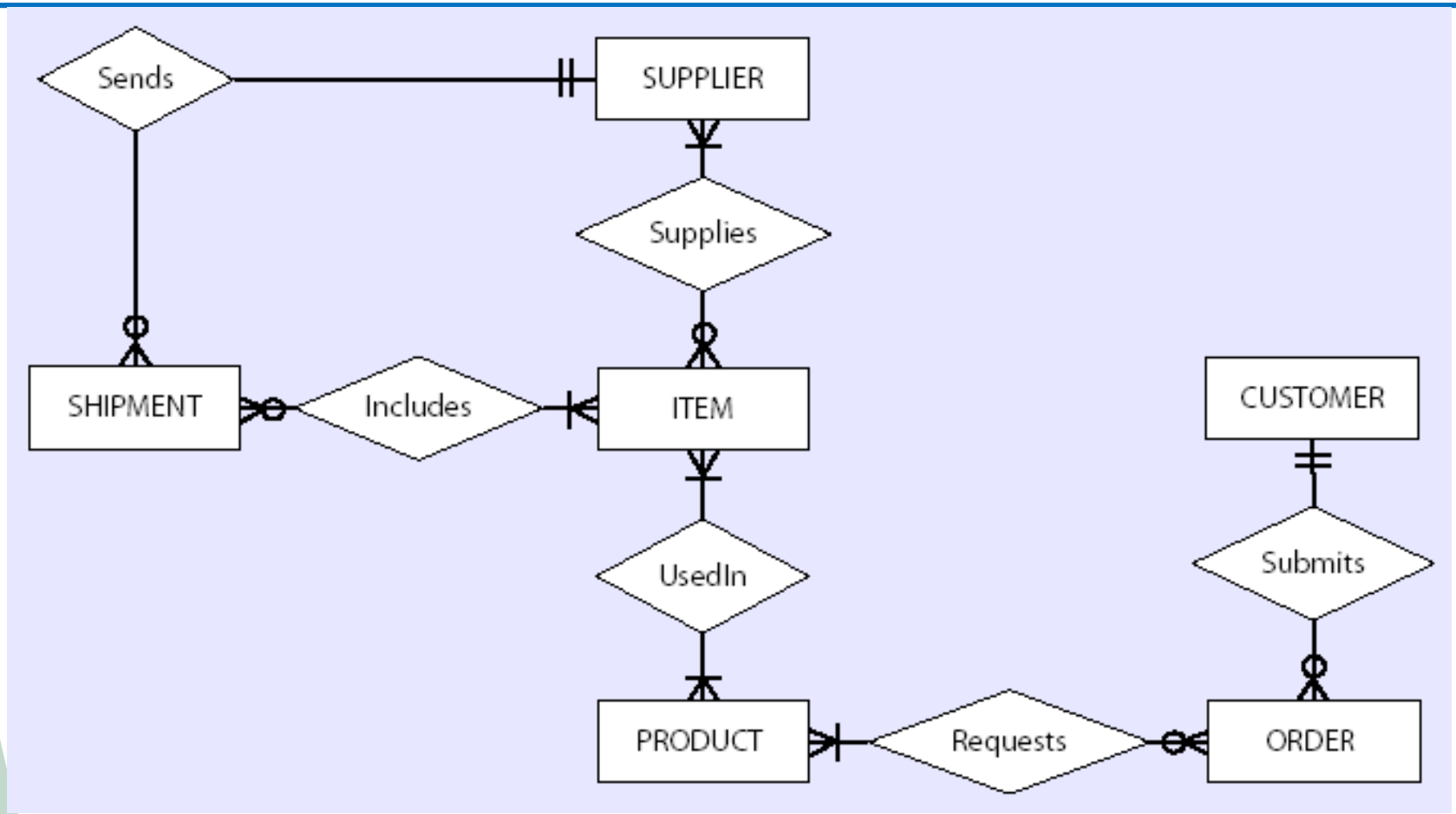


Figure 5: An Example of an E-R Diagram

How are Databases Structured?

Definition 7: Functional Dependencies

- Functional dependencies describe some of the rules that hold **between attributes** in a system.
- States whether a particular value of one data element (X) in a record **determines** a particular value of another data element (Y) for that record.
- If we know the value of X, then we can determine **a unique value** of Y.

How are Databases Structured?

- Student Number ⟶ Student Name
  - The data element Student Name is functionally dependent on Student Number.

  or

  - Student Number determines a unique value of Student Name

  or

  - Student Name is determined by Student Number
- Take note that for each Student Number there is only ONE value for Student Name

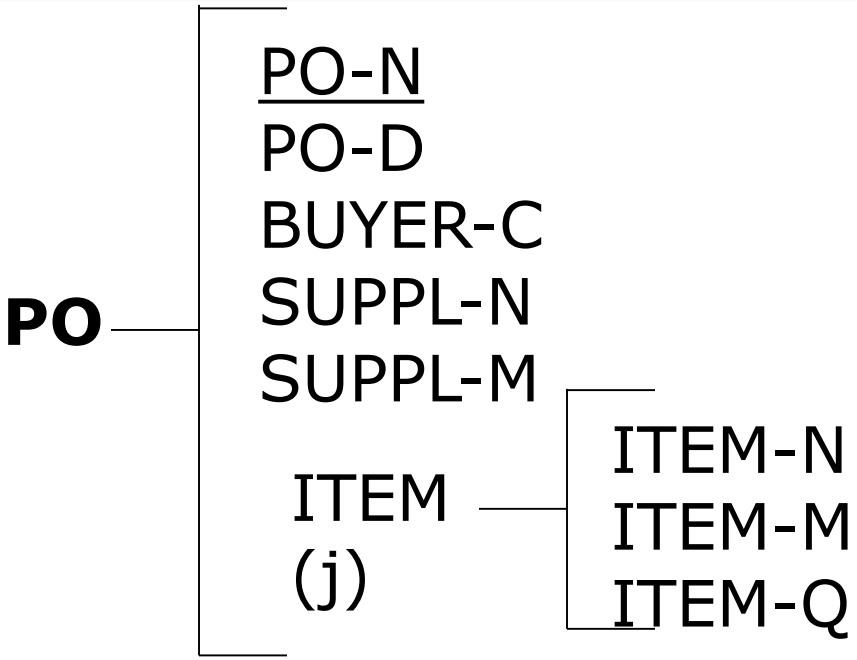
How are Databases Structured?

Definition 8: Zeroth Normal Form

- 0NF - Completely un-normalized data contains many redundancies and dependencies among its attributes as well as repeating groups or fields
- All data stores (unnormalized) are automatically assumed to be in the Zeroth Normal Form

How are Databases Structured?

Example 1: Zeroth Normal Form



PO (PO-N, PO-D, BUYER-C, SUPPL-N, SUPPL-M, (ITEM-N, ITEM-M, ITEM-Q))

How are Databases Structured?

PO-N	PO-D	BUYER-C	SUPPL-N	SUPPL-M	ITEM-N	ITEM-M	ITEM-Q
1	2/5/2009	ERLJ	335	Jabee	I02	Nut	30
1	2/5/2009	ERLJ	335	Jabee	I03	Bolt	20
1	2/5/2009	ERLJ	335	Jabee	I04	Screw	40
1	2/5/2009	ERLJ	335	Jabee	I05	Cam	20
1	2/5/2009	ERLJ	335	Jabee	I06	Cog	40
1	2/5/2009	ERLJ	335	Jabee	I07	Jig	50
2	2/6/2009	IGM	275	Mcdu	I02	Nut	30
2	2/6/2009	IGM	275	Mcdu	I03	Bolt	30
3	2/9/2009	ERLJ	223	Grinich	I04	Screw	70
3	2/9/2009	ERLJ	223	Grinich	I06	Cog	40
4	2/24/2009	EGA	230	Chiking	I03	Bolt	20
4	2/24/2009	EGA	230	Chiking	I05	Cam	30
4	2/24/2009	EGA	230	Chiking	I06	Cog	40
5	2/24/2009	VJS	208	KefC	I06	Cog	50
6	2/24/2009	ACM	335	Jabee	I08	Rivet	30

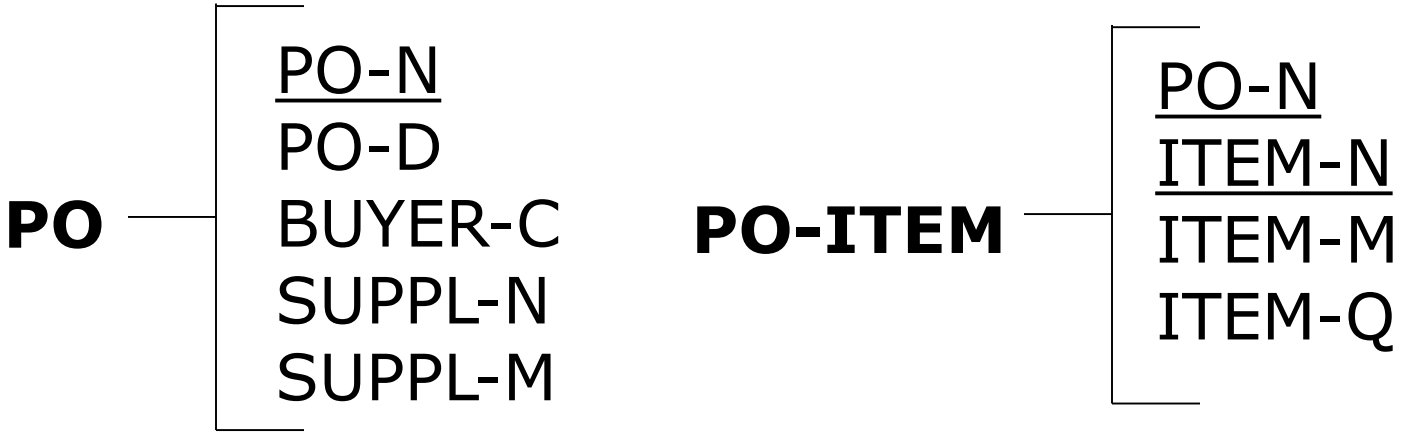
How are Databases Structured?

Definition 9: First Normal Form

- A data store is in the First Normal Form if all its data elements are atomic, that is, there are no repeating groups.
- To Transform the data store from 0NF to 1NF:
  - Move the repeating group to a new Data Store.
  - Copy the Key of the old (source) data store to the new data store
  - Determine the key of the new data store

How are Databases Structured?

Example 1 (Cont.): First Normal Form



PO (PO-N, PO-D, BUYER-C, SUPPL-N, SUPPL-M)
PO-ITEM (PO-N, ITEM-N, ITEM-M, ITEM-Q)

How are Databases Structured?

PO-N	PO-D	BUYER-C	SUPPL-N	SUPPL-M
1	02/05/09	ERLJ	335	Jabee
2	02/06/09	IGM	275	Mcdu
3	02/09/09	ERLJ	223	Grinich
4	02/24/09	EGA	230	Chiking
5	02/24/09	VJS	208	KefC
6	02/24/09	ACM	335	Jabee

PO-N	ITEM-N	ITEM-M	ITEM-Q
1	I02	Nut	30
1	I03	Bolt	20
1	I04	Screw	40
1	I05	Cam	20
1	I06	Cog	40
1	I07	Jig	50
2	I02	Nut	30
2	I03	Bolt	30
3	I04	Screw	70
3	I06	Cog	40
4	I03	Bolt	20
4	I05	Cam	30
4	I06	Cog	40
5	I06	Cog	50
6	I08	Rivet	30

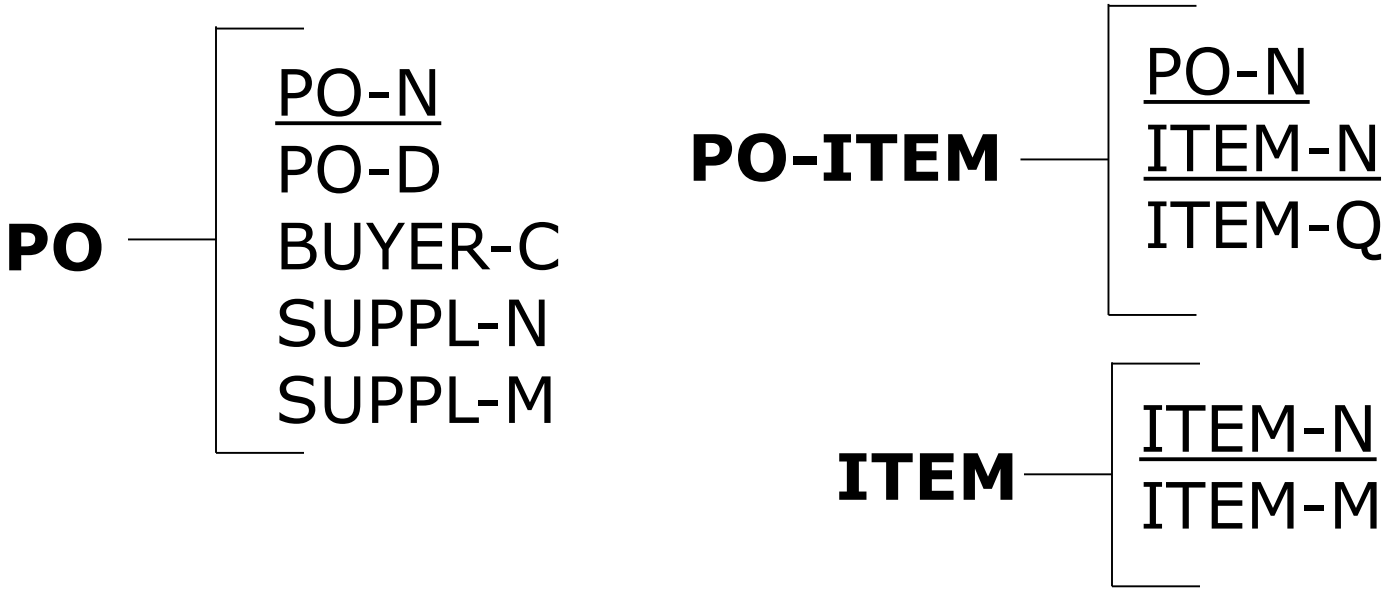
How are Databases Structured?

Definition 10: Second Normal Form

- A data store is in the Second Normal Form if:
  - It is in the First Normal Form, and
  - All the nonkey data elements are functionally dependent on the whole primary key.
- Transform the data store from 1NF to 2NF:
  - If a data element is dependent only on part of the key and not on the whole key,
    - Move the data element to a new data store
    - Copy the partial key that the data element depends on and it becomes the key of the new data store

How are Databases Structured?

Example 1 (Cont.): Second Normal Form



PO (PO-N, PO-D, BUYER-C, SUPPL-N, SUPPL-M)
PO-ITEM (PO-N, ITEM-N, ITEM-Q)
ITEM (ITEM-N, ITEM-M)

PO-N	PO-D	BUYER-C	SUPPL-N	SUPPL-M
1	02/05/09	ERLJ	335	Jabee
2	02/06/09	IGM	275	Mcdu
3	02/09/09	ERLJ	223	Grinich
4	02/24/09	EGA	230	Chiking
5	02/24/09	VJS	208	KefC
6	02/24/09	ACM	335	Jabee

PO-N	ITEM-N	ITEM-Q
1	I02	30
1	I03	20
1	I04	40
1	I05	20
1	I06	40
1	I07	50
2	I02	30
2	I03	30
3	I04	70
3	I06	40
4	I03	20
4	I05	30
4	I06	40
5	I06	50
6	I08	30

ITEM-N	ITEM-M
I02	Nut
I03	Bolt
I04	Screw
I05	Cam
I06	Cog
I07	Jig
I08	Rivet

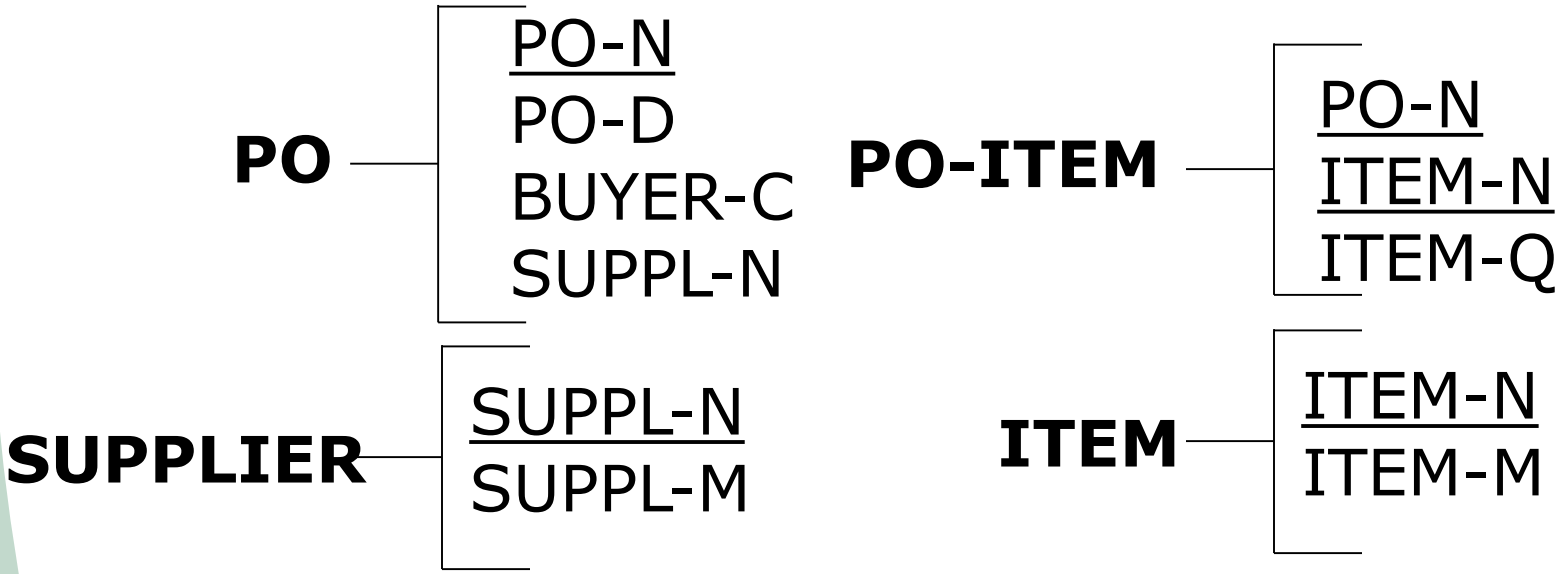
How are Databases Structured?

Definition 11: Third Normal Form

- A data store is in the Third Normal Form if:
  - It is in the Second Normal Form,
  - No nonkey data element is functionally dependent on any other nonkey data element
- Transform the data store from 2NF to 3NF:
  - If a data element is dependent on any other nonkey data element
    - Move the data element to a new data store
    - Copy the nonkey data element it depends on into this new data store
    - Determine which data element should be the key

How are Databases Structured?

Example 1 (Cont.): Third Normal Form



PO (PO-N, PO-D, BUYER-C, SUPPL-N, SUPPL-M)
PO-ITEM (PO-N, ITEM-N, ITEM-Q)
ITEM (ITEM-N, ITEM-M)
SUPPLIER (SUPPL-N, SUPPL-M)

PO-N	PO-D	BUYER-C	SUPPL-N
1	02/05/09	ERLJ	335
2	02/06/09	IGM	275
3	02/09/09	ERLJ	223
4	02/24/09	EGA	230
5	02/24/09	VJS	208
6	02/24/09	ACM	335

PO-N	ITEM-N	ITEM-Q
1	I02	30
1	I03	20
1	I04	40
1	I05	20
1	I06	40
1	I07	50
2	I02	30
2	I03	30
3	I04	70
3	I06	40
4	I03	20
4	I05	30
4	I06	40
5	I06	50
6	I08	30

SUPPL-N	SUPPL-M
208	KefC
223	Grinich
230	Chiking
275	Mcdu
335	Jabee

ITEM-N	ITEM-M
I02	Nut
I03	Bolt
I04	Screw
I05	Cam
I06	Cog
I07	Jig
I08	Rivet

Outline for this Session

- What is a Database?
- How are databases structured?
- **How do we manipulate data in a database?**

eljalao@up.edu.ph

SQL

Definition 12: SQL

- SQL: Structured Query Language
  - Programming language designed for managing data in a RDBMS
- Why SQL?
  - The definition of a relational system, requires that a single language
  - Able to handle all communications with the database.

SQL

- SQL statement consists of reserved words and user-defined words.
- Reserved words are a fixed part of SQL and must be spelt exactly as required
- User-defined words are made up by user and represent names of various database objects such as relations, columns, views.

```
CREATE TABLE Staff(staffNo VARCHAR(5),
      lName VARCHAR(15),
      salary DECIMAL(7,2));
```

SQL

Example 2: SQL

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000.00	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000.00	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000.00	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000.00	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000.00	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000.00	B005

```
SELECT staffNo, fName, lName, position, sex,
       DOB,  salary, branchNo
FROM Staff;


SELECT * FROM Staff;
```

SQL

- Select All Columns

```
SELECT staffNo, fName, lName, position, sex,
DOB,   salary, branchNo
FROM Staff;
```

  - or

```
SELECT * FROM Staff;
```

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000.00	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000.00	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000.00	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000.00	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000.00	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000.00	B005

SQL

- Select Specific Columns

```
SELECT staffNo, fName, lName, salary
FROM Staff;
```

- Result

staffNo	fName	lName	salary
SL21	John	White	30000.00
SG37	Ann	Beech	12000.00
SG14	David	Ford	18000.00
SA9	Mary	Howe	9000.00
SG5	Susan	Brand	24000.00
SL41	Julie	Lee	9000.00

SQL

- List all staff with a salary greater than 10,000.

```
SELECT staffNo, fName, lName, position, salary
FROM Staff
WHERE salary > 10000;
```

staffNo	fName	lName	position	salary
SL21	John	White	Manager	30000.00
SG37	Ann	Beech	Assistant	12000.00
SG14	David	Ford	Supervisor	18000.00
SG5	Susan	Brand	Manager	24000.00

SQL

- SELECT Statement - Aggregates
  - ISO standard defines five aggregate functions:
    - COUNT returns number of values in specified column.
    - SUM returns sum of values in specified column.
    - AVG returns average of values in specified column.
    - MIN returns smallest value in specified column.
    - MAX returns largest value in specified column.

SQL

- SELECT Statement - Aggregates
  - Each operates on a **single column** of a table and returns a single value.
  - COUNT, MIN, and MAX apply to **numeric and non-numeric** fields, but SUM and AVG may be used on numeric fields only.
  - Apart from COUNT(*), each function eliminates nulls first and operates only on remaining non-null values.
  - COUNT(*) **counts all rows** of a table, regardless of whether nulls or duplicate values occur.
  - Can use DISTINCT before column name to eliminate **duplicates.**
  - DISTINCT has **no effect** with MIN/MAX, but may have with SUM/AVG.

SQL

- How many properties cost more than £350 per month to rent from PropertyForRent relation?
- PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo, staffNo, branchNo)

```
SELECT COUNT(*) AS count
FROM PropertyForRent
WHERE rent > 350;
```

count
5

SQL

- How many different properties were viewed (viewdate) for the entire month of May ’01 from Viewing elation?

- Viewing (clientNo, propertyNo, viewDate, comment)

count
2

```
SELECT COUNT(DISTINCT propertyNo) AS count
 FROM Viewing WHERE viewDate BETWEEN ‘1-May-01’
 AND ‘31-May-01’;
```

SQL

- Find number of Managers and sum of their salaries from Staff relation.

count	sum
2	54000.00

- Staff (staffNo, fName, IName, position, sex, DOB, salary, branchNo)

```
SELECT COUNT(staffNo) AS count,
       SUM(salary) AS sum
FROM Staff
WHERE position = 'Manager';
```

SQL

- SELECT Statement - Grouping
  - Use GROUP BY clause to get **sub-totals.**
  - SELECT and GROUP BY closely integrated: each item in SELECT list must be *single-valued per group*, and SELECT clause may only contain:
    - column names
    - aggregate functions
    - constants
    - expression involving combinations of the above.
  - All column names in SELECT list **must appear** in GROUP BY clause unless name is used only in an aggregate function.
  - If WHERE is used with GROUP BY, WHERE is applied first, then groups are formed from remaining rows **satisfying the predicate**.
  - ISO considers two nulls to be **equal** for purposes of GROUP BY.

SQL

- Find number of staff in each branch and their total salaries and sort it by branchNo.
- Staff (staffNo, fName, lName, position, sex, DOB, salary, branchNo)

branchNo	count	sum
B003	3	54000.00
B005	2	39000.00
B007	1	9000.00

```
SELECT branchNo,COUNT(staffNo) AS count,
SUM(salary) AS sum
FROM Staff GROUP BY branchNo
ORDER BY branchNo;
```

SQL

- Joins: List names of all clients who have viewed a property along with any comment supplied.

Client

clientNo	fName	lName	telNo	prefType	maxRent
CR76	John	Kay	0207-774-5632	Flat	425
CR56	Aline	Stewart	0141-848-1825	Flat	350
CR74	Mike	Ritchie	01475-392178	House	750
CR62	Mary	Tregear	01224-196720	Flat	600

Viewing

clientNo	propertyNo	viewDate	comment
CR56	PA14	24-May-04	too small
CR76	PG4	20-Apr-04	too remote
CR56	PG4	26-May-04	
CR62	PA14	14-May-04	no dining room
CR56	PG36	28-Apr-04	

clientNo	fName	lName	propertyNo	comment
CR56	Aline	Stewart	PG36	
CR56	Aline	Stewart	PA14	too small
CR56	Aline	Stewart	PG4	
CR62	Mary	Tregear	PA14	no dining room
CR76	John	Kay	PG4	too remote

SQL

```
SELECT c.clientNo, fName, lName,
propertyNo, comment
FROM Client c
INNER JOIN Viewing v
ON c.clientNo = v.clientNo;
```

SQL

- Insert a new row into Staff table supplying data for all columns: (‘SG16’, ‘Alan’, ‘Brown’, ‘Assistant’, ‘M’, Date‘1957-05-25’, 8300, ‘B003’);

```
INSERT INTO Staff
VALUES  (‘SG16’,  ‘Alan’,  ‘Brown’,  ‘Assistant’,
   ‘M’, Date‘1957-05-25’, 8300, ‘B003’);
```

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000.00	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000.00	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000.00	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000.00	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000.00	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000.00	B005

SQL

- Promote David Ford (staffNo='SG14') to Manager and change his salary to £18,000.

```
UPDATE Staff
SET position = 'Manager', salary = 18000
WHERE staffNo = 'SG14';
```

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000.00	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000.00	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000.00	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000.00	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000.00	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000.00	B005

Outline for this Session

- What is a Database?
- How are databases structured?
- How do we manipulate data in a database?

eljalao@up.edu.ph

References

- www.cs.utexas.edu/~mitra/csFall2011/cs329/lectures/SQL.ppt