MIS781 Business Intelligence and Database

Module 3

Introduction to OLTP Database

EFMD EQUIS ACCREDITED

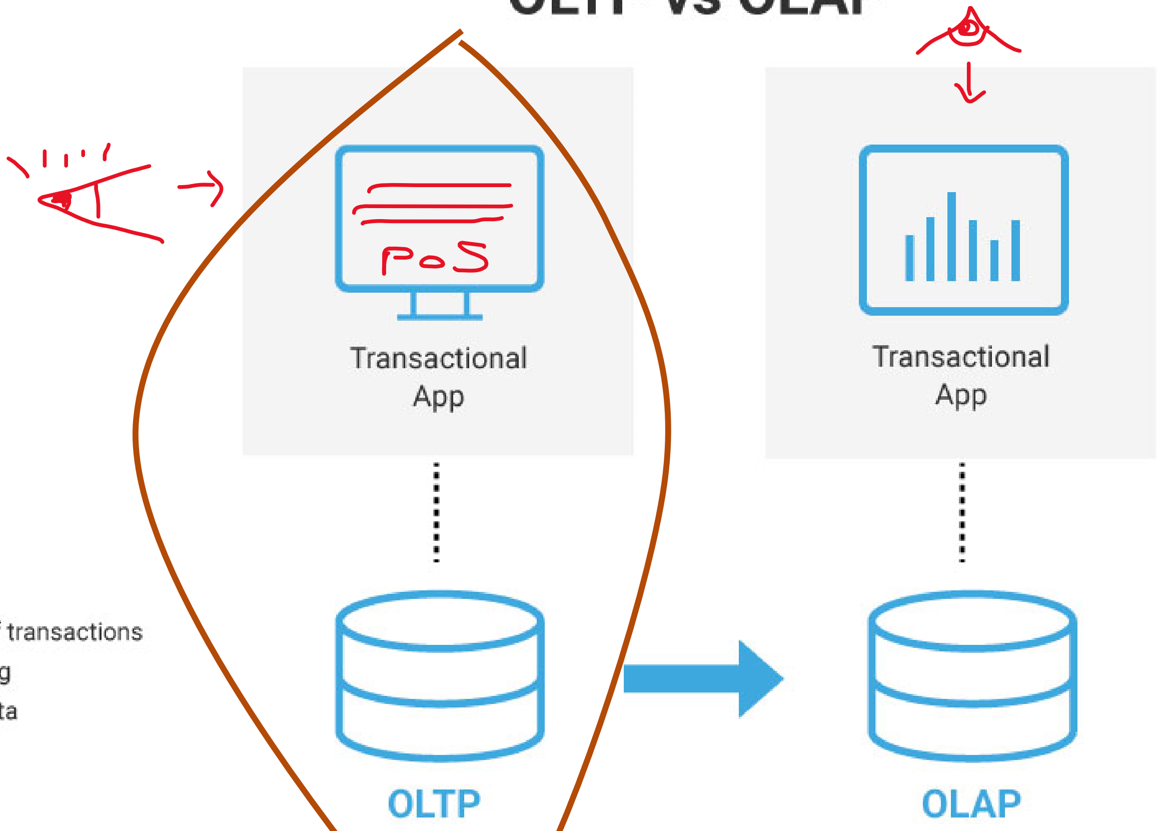
AACSB ACCREDITED

DEAKIN BUSINESS SCHOOL

Business Analyst memes by IIBA
What My Clients Think I Do
What My Co-workers Think I Do
What My Parents Think I Do
What My Friends Think I Do
What I Think I Do
What I Really Do

Online Transaction Processing OLTP vs Online Analytical Processing OLAP

OLTP vs OLAP



- High volume of transactions
- Fast Processing
- Normalized Data
- Many Tables

«Who bought X?»

- High volume of data
- Slow Queries
- Denormalized Data
- Fewer Tables

«How many people bought X?»

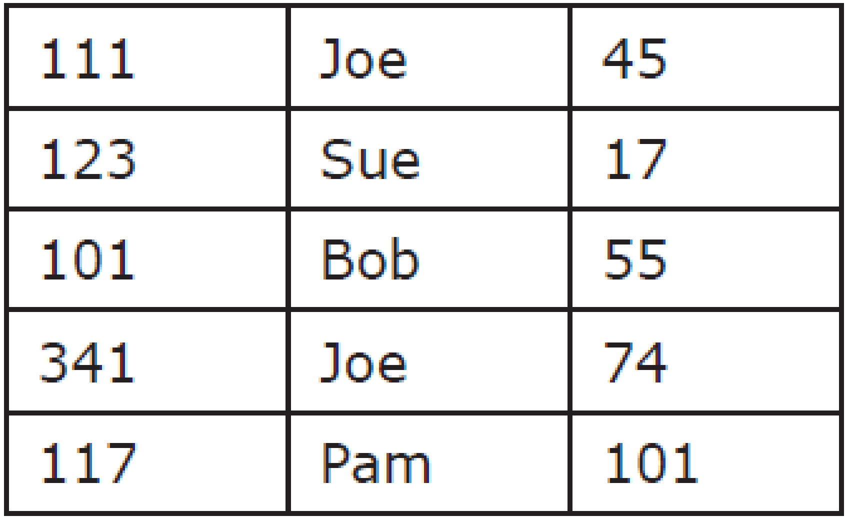
Source: Jukic et al. (2020) Database Systems - Introduction to Databases and Data Warehouses, Prospect Press.

OPERATIONAL (OLTP) vs ANALYTICAL (OLAP) DATABASES

- **Operational information (transactional information)** - the information collected and used in support of day to day operational needs in businesses and other organisations
- **Operational database** - collects and presents operational information in support of daily operational procedures and processes
- **Analytical information** - the information collected and used in support of analytical tasks
    - Analytical information is based on operational (transactional) information
- **Analytical database** - collects and presents analytical information in support of analytical tasks

TERMINOLOGY

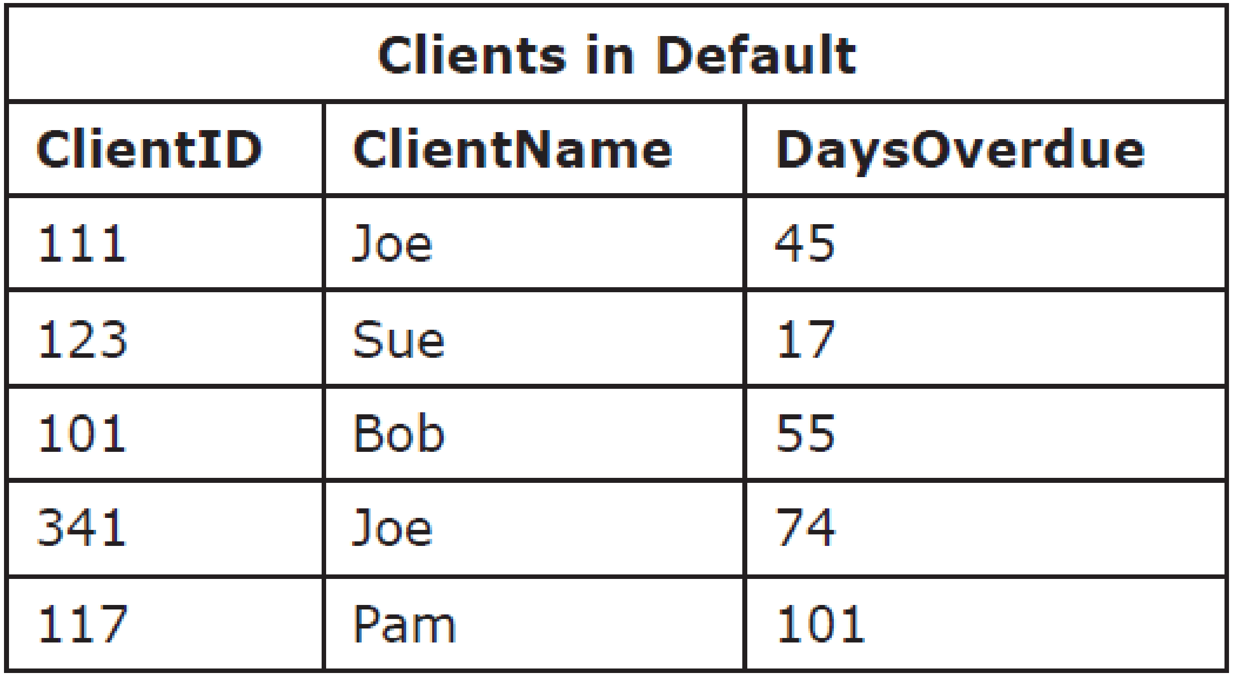
Data without metadata - example



111	Joe	45
123	Sue	17
101	Bob	55
341	Joe	74
117	Pam	101

TERMINOLOGY

Data with metadata - example



Clients in Default		
ClientID	ClientName	DaysOverdue
111	Joe	45
123	Sue	17
101	Bob	55
341	Joe	74
117	Pam	101

IMPORTANT TERMINOLOGY

- **Data** - facts that are recorded and can be accessed
  - Data formats – text, numbers, figures, graphics, images, audio/video recordings and more
  - Data is recorded and kept because it is considered to be of use to an intended user
- **Information** - refers to the data that is accessed by a user for some particular purpose
  - Typically, getting the needed information from a collection of data requires performing an activity, such as searching through, processing, or manipulating the data in some form or fashion
- **Metadata** - data that describes the structure and the properties of the data (see next slide)
  - Metadata is essential for the proper understanding and use of the data

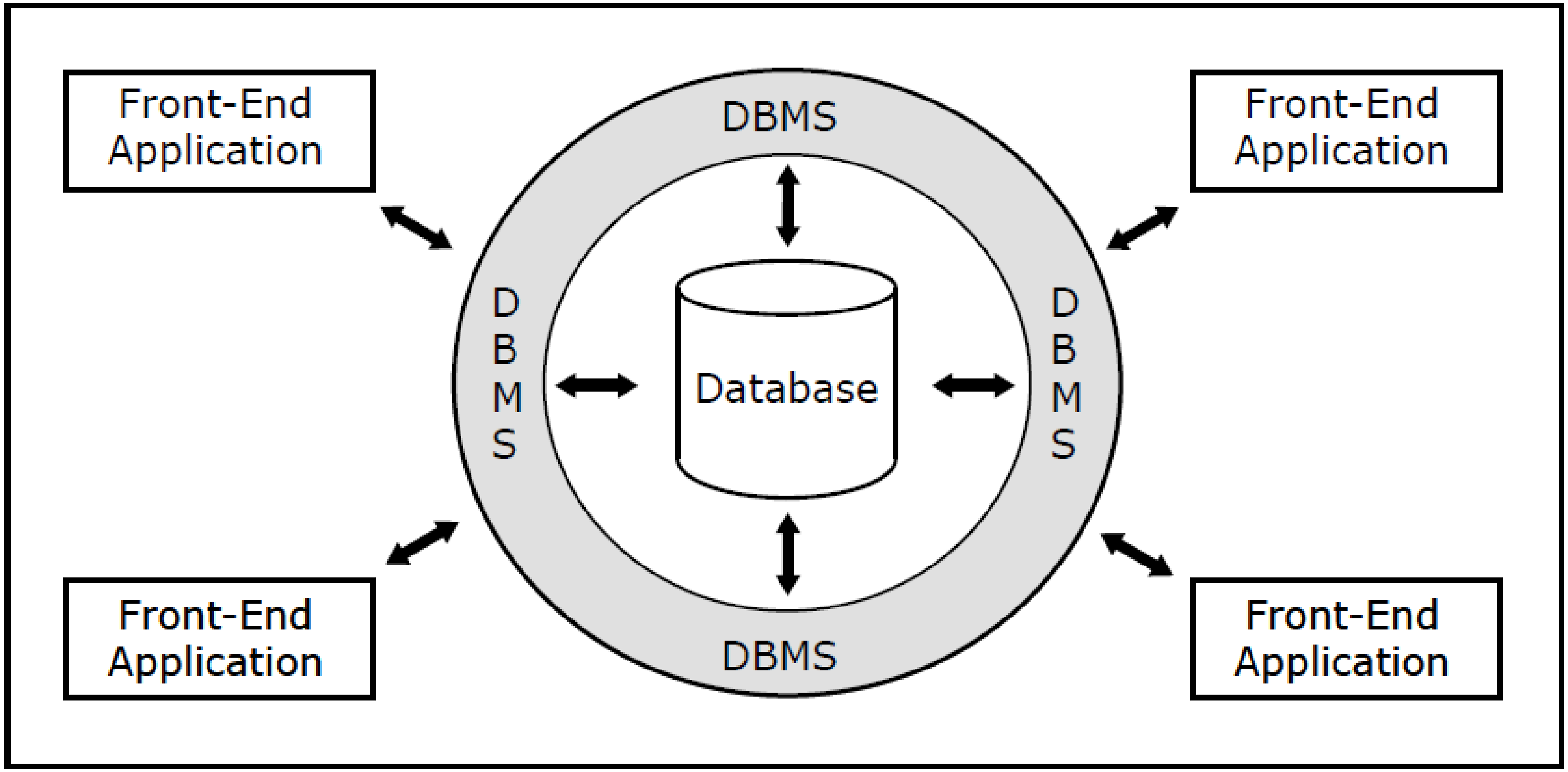
TERMINOLOGY

- **Database management system (DBMS)** - software used for:
  - Creation of databases
  - Insertion, storage, retrieval, update, and deletion of the data in the database
  - Maintenance of databases (see next slide)

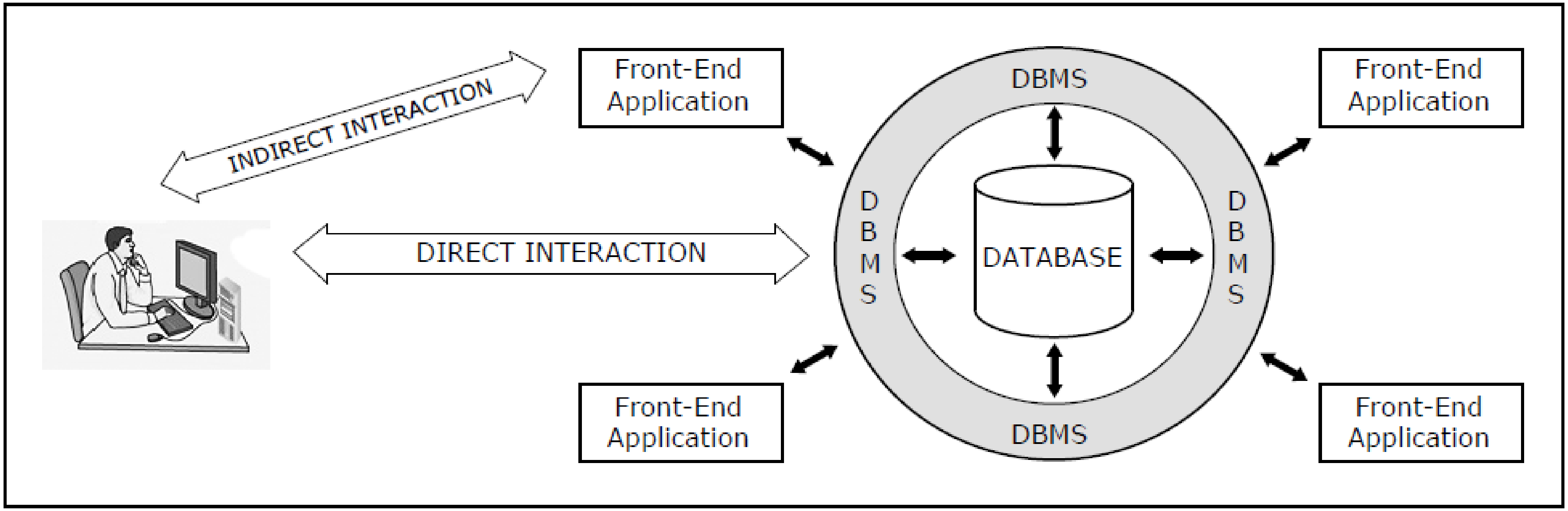
- **Business rules** - the conditions that define how the business operates and should be analysed. E.g. An international student at Deakin must be a full-time student, hence the student must enroll to at least 3 units per semester.

TERMINOLOGY

Typical database system architecture



TERMINOLOGY: Direct vs. Indirect Interaction



TERMINOLOGY

- **Front-end applications** - provide a mechanism for easy interaction between the users and the DBMS
- **End-users** (**business-users**) - users using a database system to support their tasks and processes
- **Indirect interaction** - end-user communicating with the database through front-end applications
- **Direct interaction** - end-user communicating with the database directly through DBMS

Relational Database Modeling (RDM)

- **Relational database model -** logical database model that represents a database as a collection of related tables
- Most contemporary commercial DBMS software packages, are **relational DBMS (RDBMS)** software packages

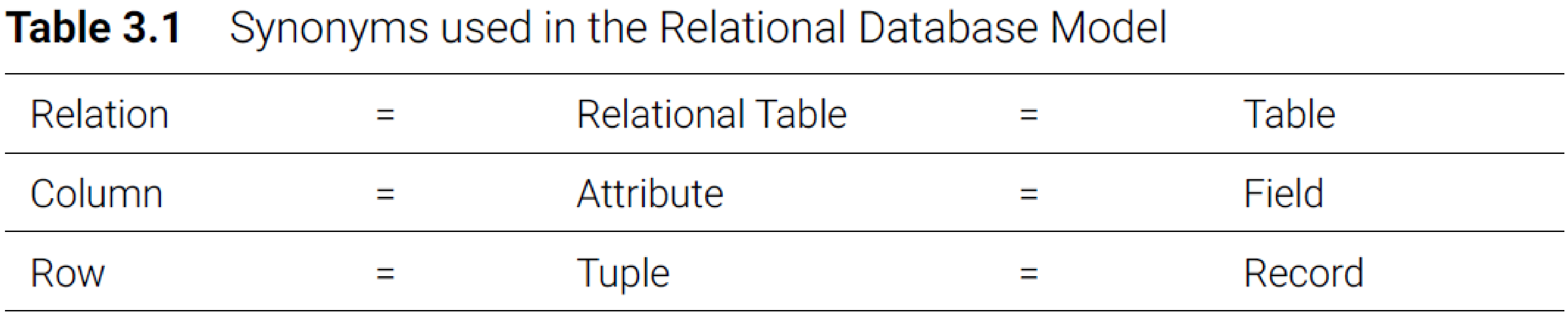
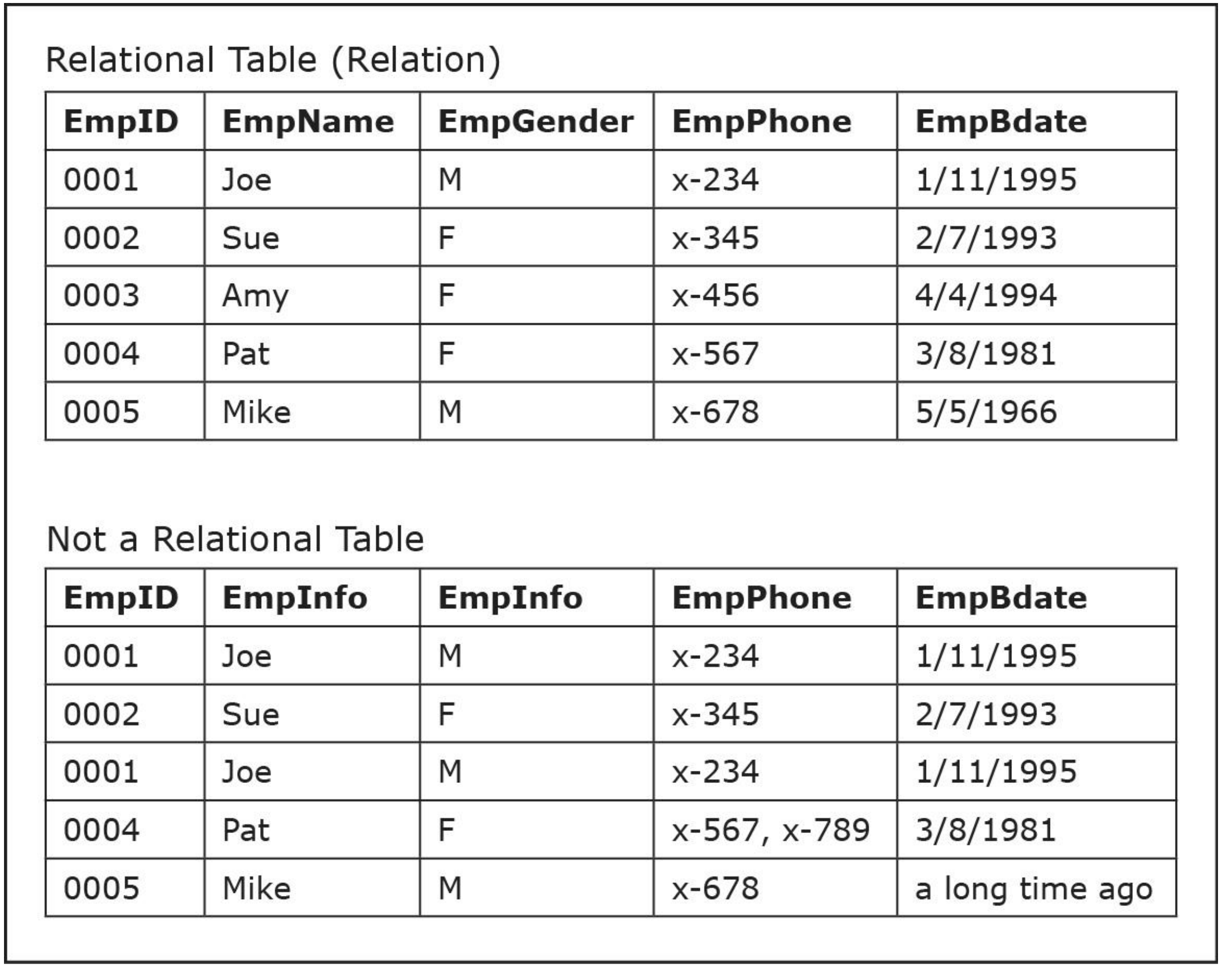


Table 3.1 Synonyms used in the Relational Database Model

Relation	=	Relational Table	=	Table
Column	=	Attribute	=	Field
Row	=	Tuple	=	Record

RDM: INTRODUCTION

Example of relational and non-relational tables



Relational Table (Relation)

EmpID	EmpName	EmpGender	EmpPhone	EmpBdate
0001	Joe	M	x-234	1/11/1995
0002	Sue	F	x-345	2/7/1993
0003	Amy	F	x-456	4/4/1994
0004	Pat	F	x-567	3/8/1981
0005	Mike	M	x-678	5/5/1966

Not a Relational Table

EmpID	EmpInfo	EmpInfo	EmpPhone	EmpBdate
0001	Joe	M	x-234	1/11/1995
0002	Sue	F	x-345	2/7/1993
0001	Joe	M	x-234	1/11/1995
0004	Pat	F	x-567, x-789	3/8/1981
0005	Mike	M	x-678	a long time ago

RDM INTRODUCTION

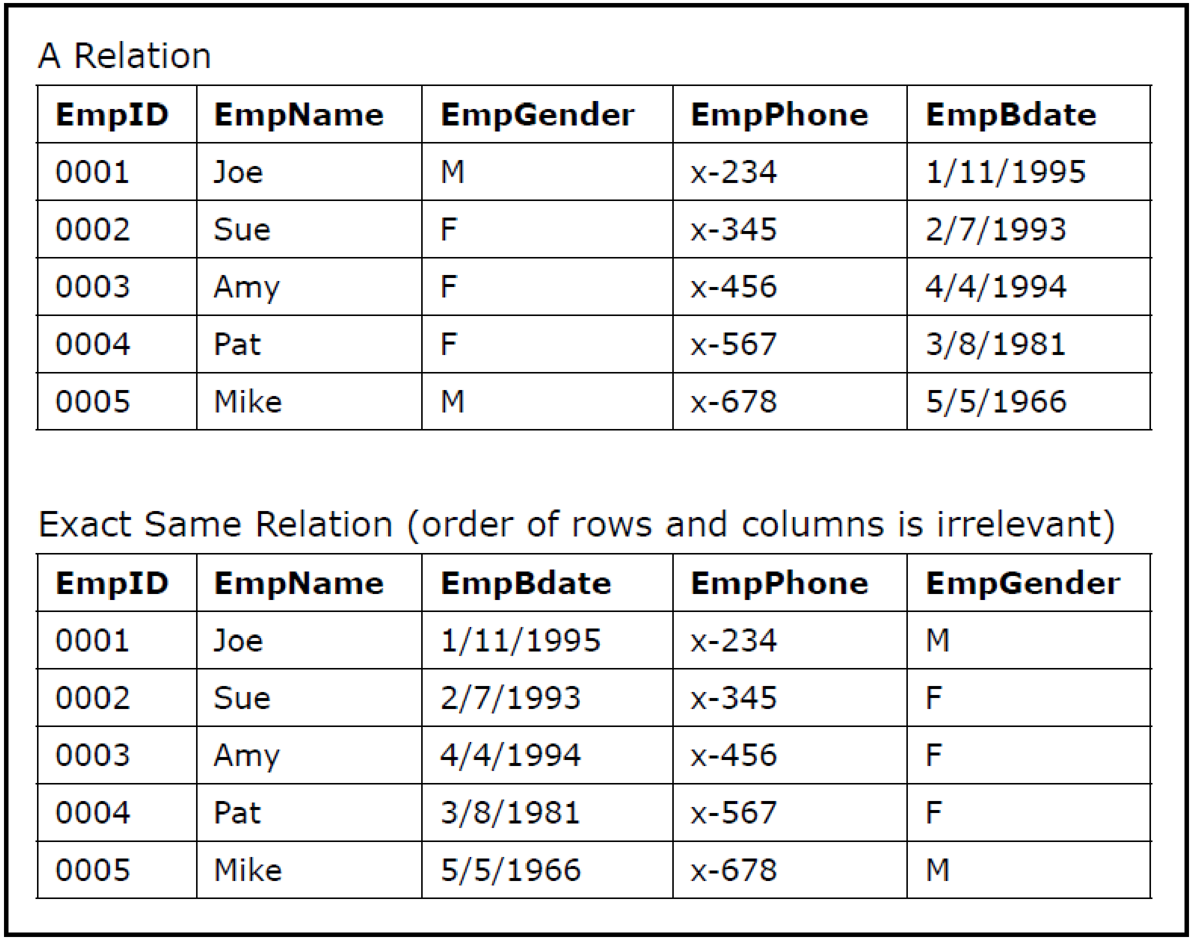
- **Relation** - table in a relational database
  - A table containing rows and columns
  - The main construct in the relational database model
  - In order for a table to be a relation the following conditions must hold:
    - ***Within one table, each column must have a unique name.***
    - ***Within one table, each row must be unique.***
    - ***All values in each column must be from the same (predefined) domain.***
    - ***Within each row, each value in each column must be single valued*** *(one value from a predefined domain, within each row in each column). See next slide example*

RDM: INTRODUCTION

- **Relation** - table in a relational database
  - Two additional properties of each table:
    - *Order of columns is irrelevant.*
    - *Order of rows is irrelevant.*

RDM: INTRODUCTION

Example of a relation with rows and columns appearing in a different order



A Relation

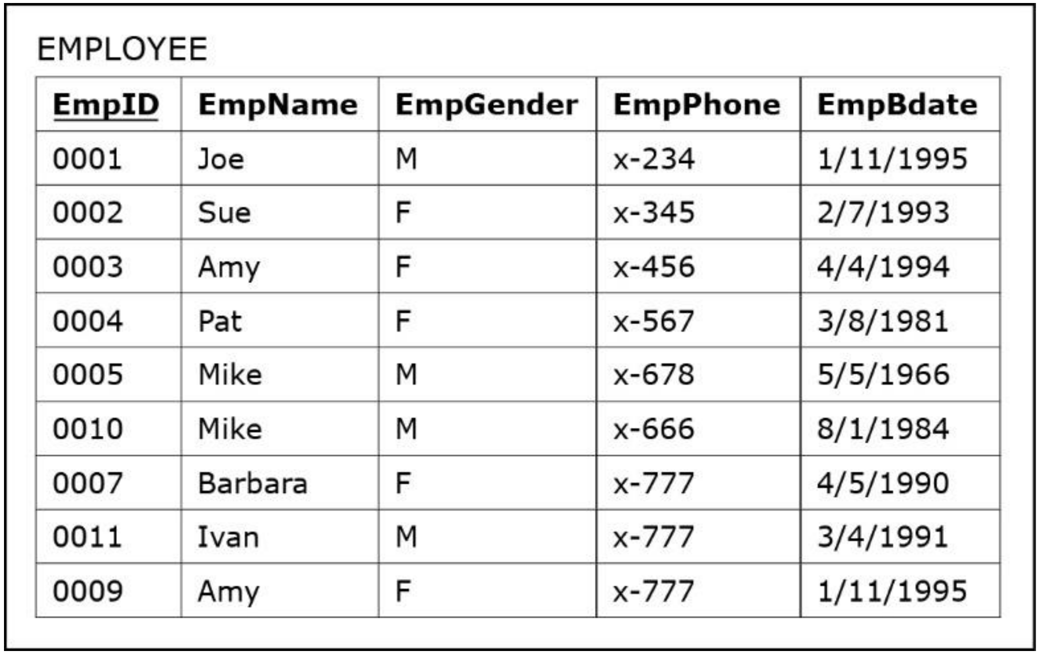
EmpID	EmpName	EmpGender	EmpPhone	EmpBdate
0001	Joe	M	x-234	1/11/1995
0002	Sue	F	x-345	2/7/1993
0003	Amy	F	x-456	4/4/1994
0004	Pat	F	x-567	3/8/1981
0005	Mike	M	x-678	5/5/1966

Exact Same Relation (order of rows and columns is irrelevant)

EmpID	EmpName	EmpBdate	EmpPhone	EmpGender
0001	Joe	1/11/1995	x-234	M
0002	Sue	2/7/1993	x-345	F
0003	Amy	4/4/1994	x-456	F
0004	Pat	3/8/1981	x-567	F
0005	Mike	5/5/1966	x-678	M

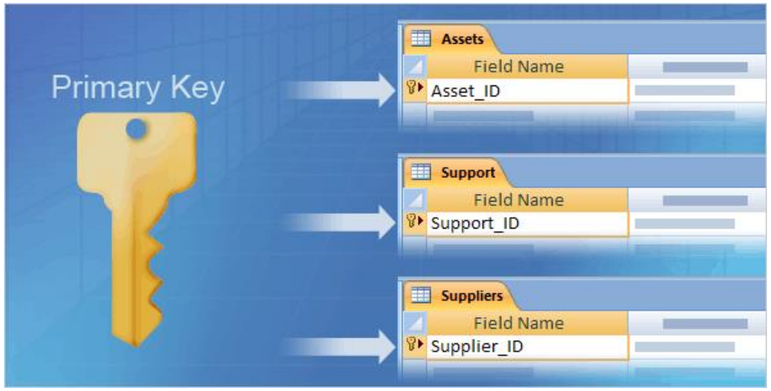
RDM: INTRODUCTION

- **Primary key** – to distinguish one row from another in a relation
  - Each relation must have a primary key
  - The name of the primary key column is **underlined** in order to distinguish it from the other columns in the relation



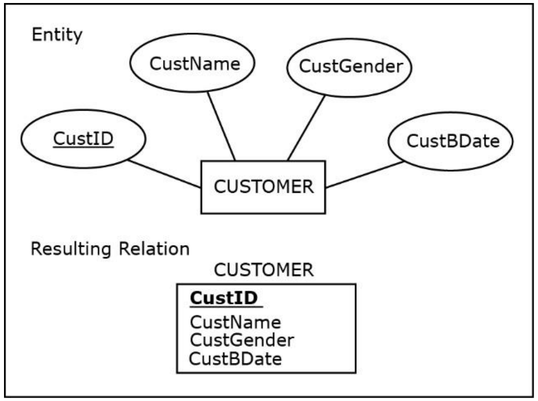
EMPLOYEE

EmpID	EmpName	EmpGender	EmpPhone	EmpBdate
0001	Joe	M	x-234	1/11/1995
0002	Sue	F	x-345	2/7/1993
0003	Amy	F	x-456	4/4/1994
0004	Pat	F	x-567	3/8/1981
0005	Mike	M	x-678	5/5/1966
0010	Mike	M	x-666	8/1/1984
0007	Barbara	F	x-777	4/5/1990
0011	Ivan	M	x-777	3/4/1991
0009	Amy	F	x-777	1/11/1995



MAPPING ENTITIES

- Mapping entities into relations
  - Each regular entity becomes a relation
  - Each regular attribute of a regular entity becomes a column of the newly created relation
  - If **an entity has a single unique attribute**, then **that attribute becomes the primary key** in the resulting mapped relation



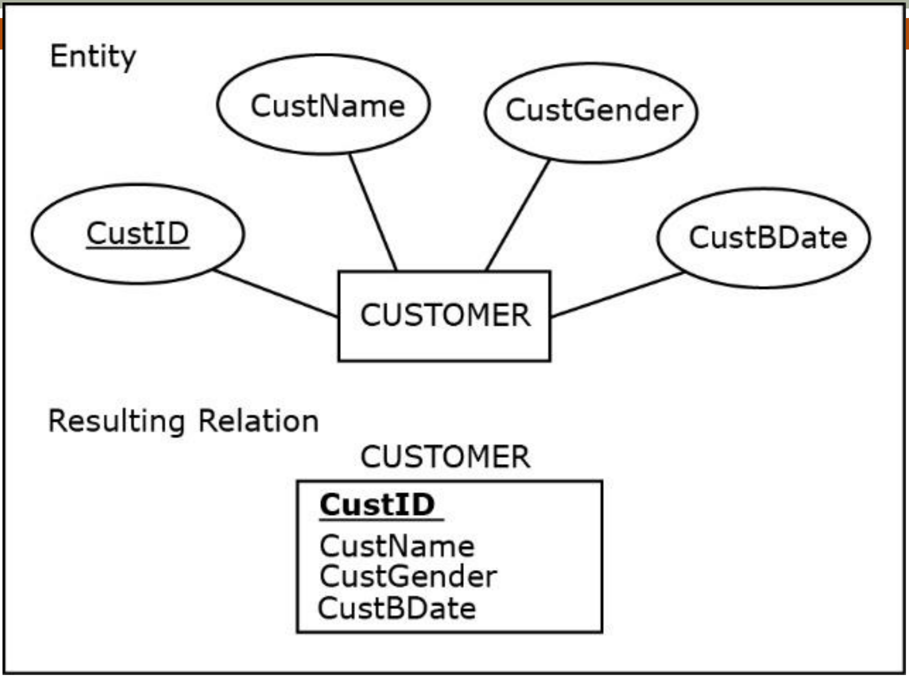
Resulting Relation

CUSTOMER

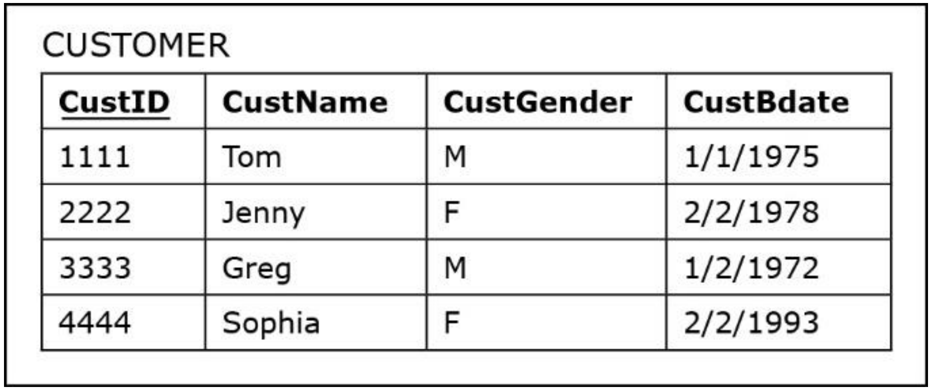
CustID
CustName
CustGender
CustBDate

MAPPING ENTITIES

Entity mapped into a relation



Sample data records for the mapped relation

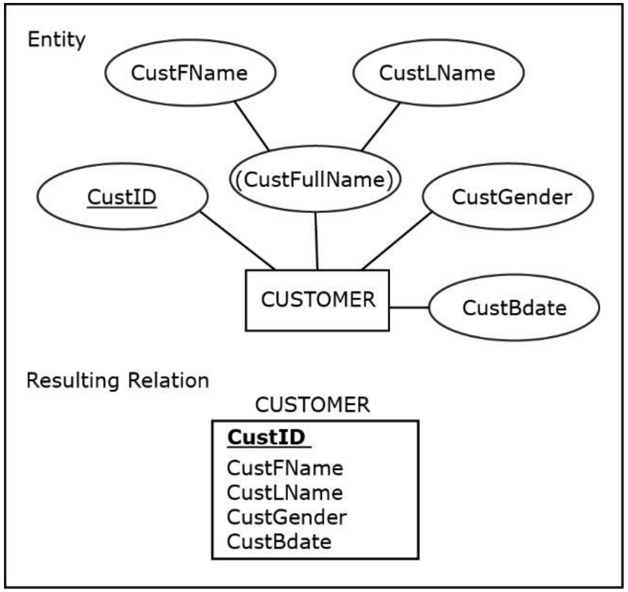


CUSTOMER

CustID	CustName	CustGender	CustBdate
1111	Tom	M	1/1/1975
2222	Jenny	F	2/2/1978
3333	Greg	M	1/2/1972
4444	Sophia	F	2/2/1993

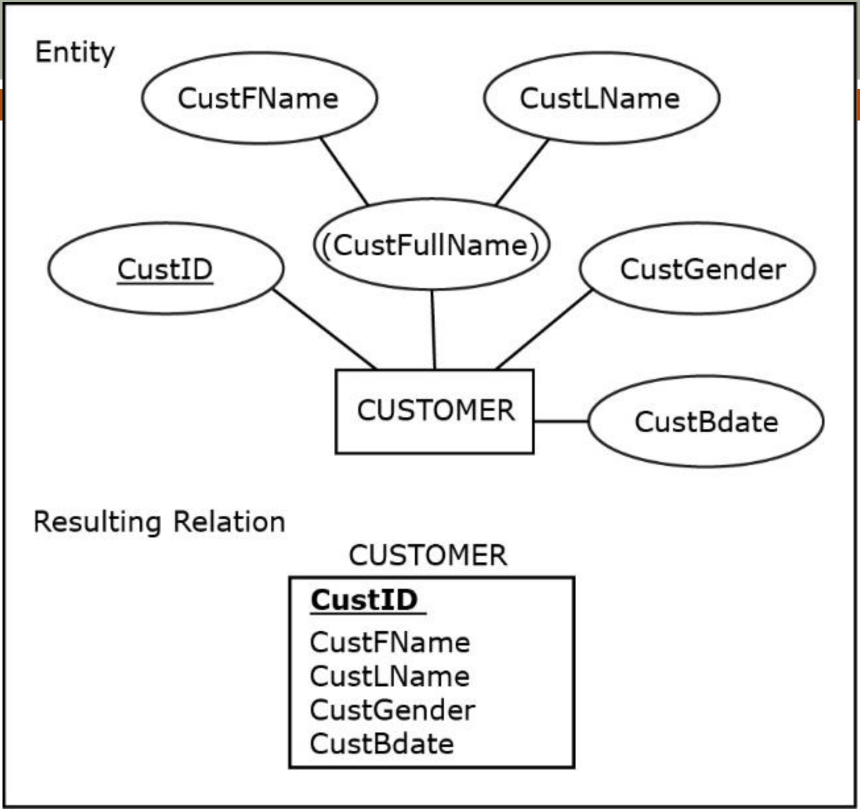
MAPPING ENTITIES WITH COMPOSITE ATTRIBUTES

- **Mapping entities with composite attributes into relations**
  - Each component of a composite attribute is mapped as a column of a relation
  - The composite attribute itself does not appear in the mapped relation



MAPPING ENTITIES WITH COMPOSITE ATTRIBUTES

Entity with a composite attribute mapped into a relation

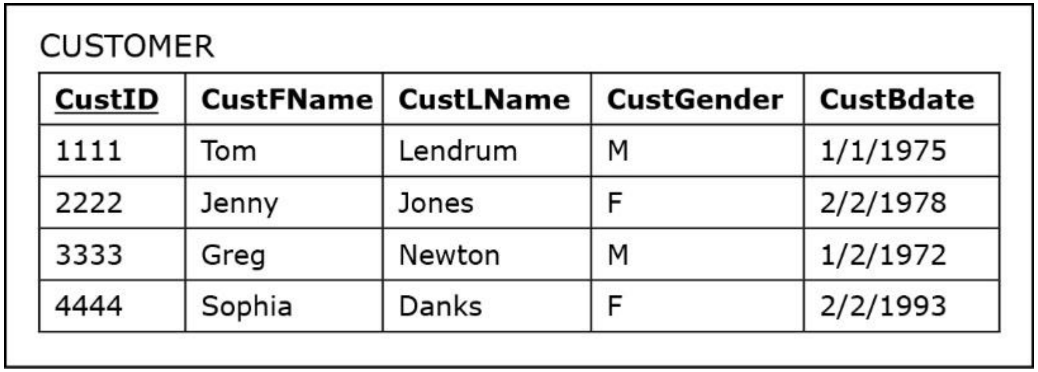


Resulting Relation

CUSTOMER

CustID
CustFName
CustLName
CustGender
CustBdate

Sample data records for the mapped relation

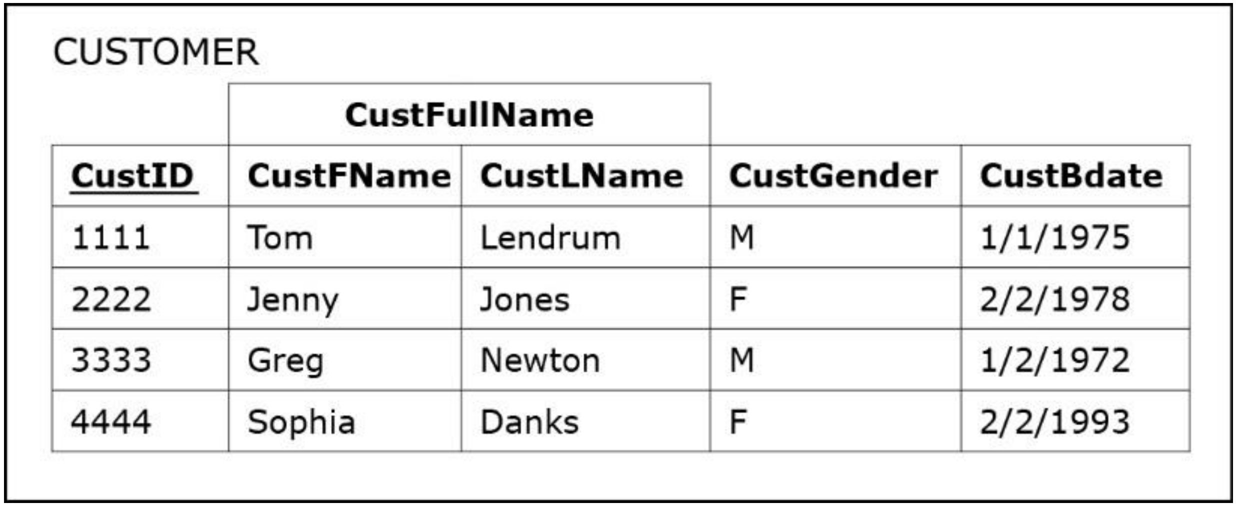


CUSTOMER

CustID	**CustFName**	**CustLName**	**CustGender**	**CustBdate**
1111	Tom	Lendrum	M	1/1/1975
2222	Jenny	Jones	F	2/2/1978
3333	Greg	Newton	M	1/2/1972
4444	Sophia	Danks	F	2/2/1993

MAPPING ENTITIES WITH COMPOSITE ATTRIBUTES

The mapped relation as presented to a user in a **front-end application**



CUSTOMER

	CustFullName			
CustID	**CustFName**	**CustLName**	**CustGender**	**CustBdate**
1111	Tom	Lendrum	M	1/1/1975
2222	Jenny	Jones	F	2/2/1978
3333	Greg	Newton	M	1/2/1972
4444	Sophia	Danks	F	2/2/1993

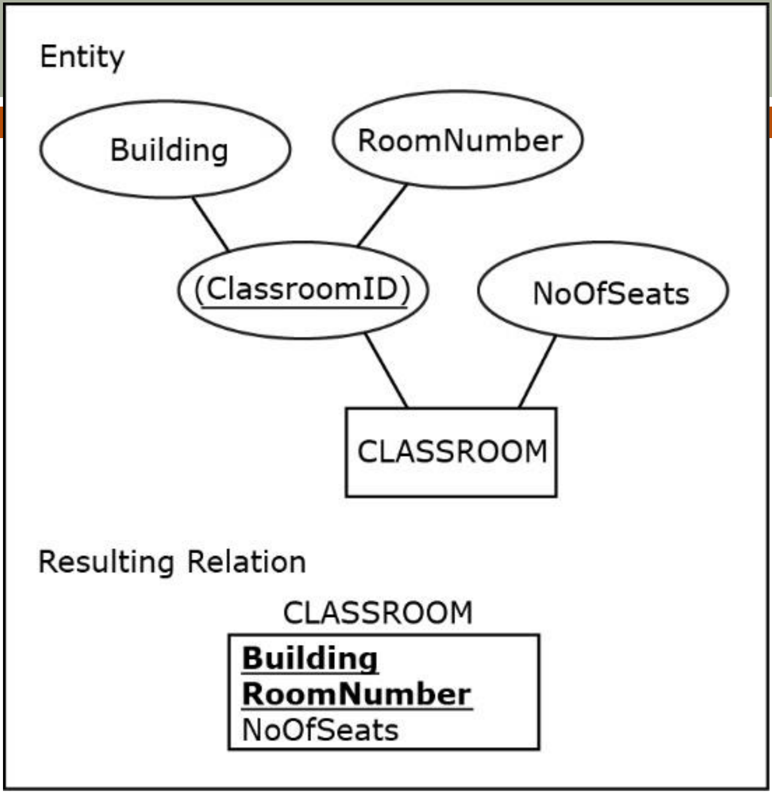
COMPOSITE PRIMARY KEY

- **Composite primary key** - a primary key that is composed of multiple columns
  - Column names of a composite primary key are underlined, because **combined together they form the primary key**

- **Mapping entities with unique composite attributes into relations**
  - An entity whose only unique attribute is a composite attribute is mapped as a relation with a composite primary key

MAPPING ENTITIES WITH UNIQUE COMPOSITE ATTRIBUTES

Entity with a unique composite attribute mapped into a relation

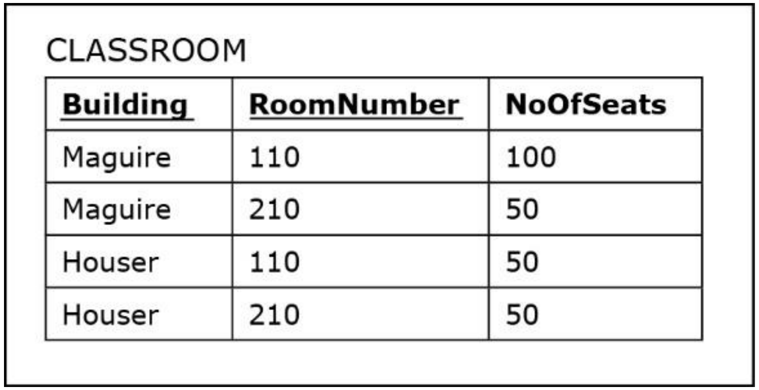


Resulting Relation

CLASSROOM

Building
RoomNumber
NoOfSeats

Sample data records for the mapped relation

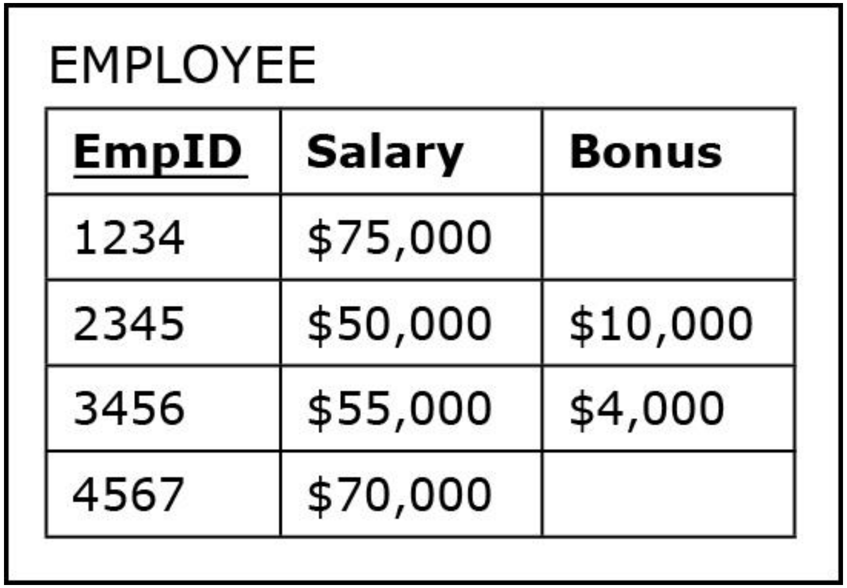


CLASSROOM

Building	**RoomNumber**	**NoOfSeats**
Maguire	110	100
Maguire	210	50
Houser	110	50
Houser	210	50

MAPPING ENTITIES WITH OPTIONAL ATTRIBUTES

- Mapping entities with optional attributes into relations
  - Optional attribute of an entity is mapped as an optional column

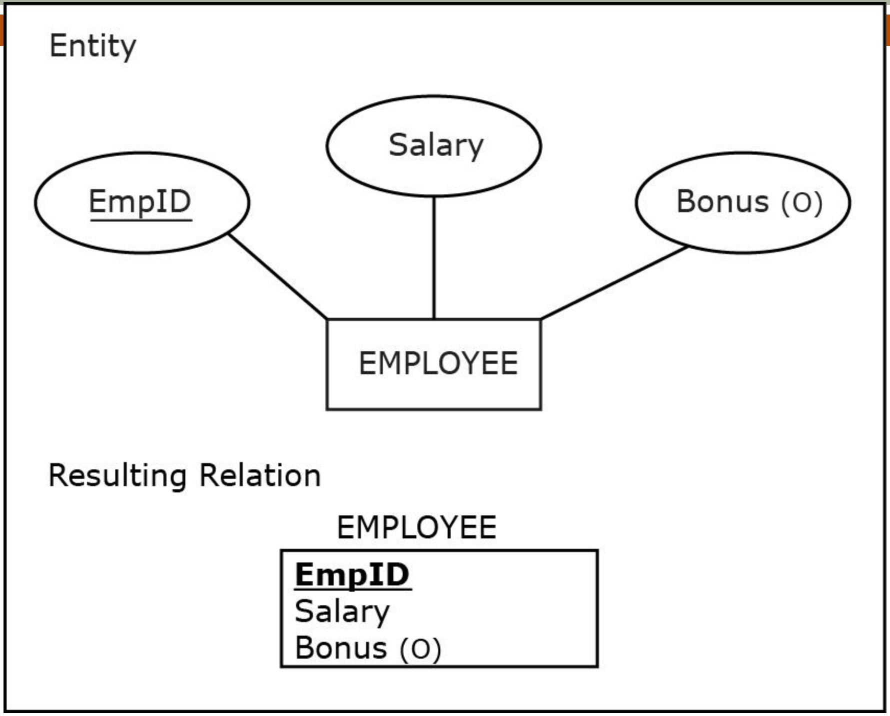


EMPLOYEE

EmpID	**Salary**	**Bonus**
1234	$75,000	
2345	$50,000	$10,000
3456	$55,000	$4,000
4567	$70,000	

MAPPING ENTITIES WITH OPTIONAL ATTRIBUTES

Entity with an optional attribute mapped into a relation

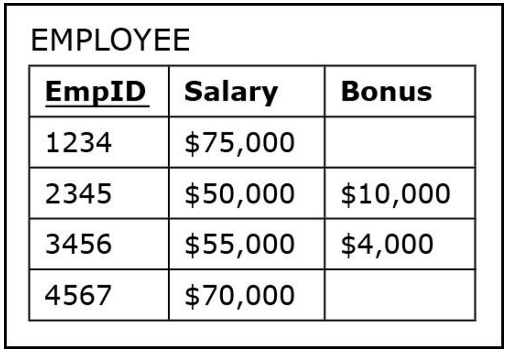


Resulting Relation

EMPLOYEE

EmpID
Salary
Bonus (O)

Sample data records for the mapped relation

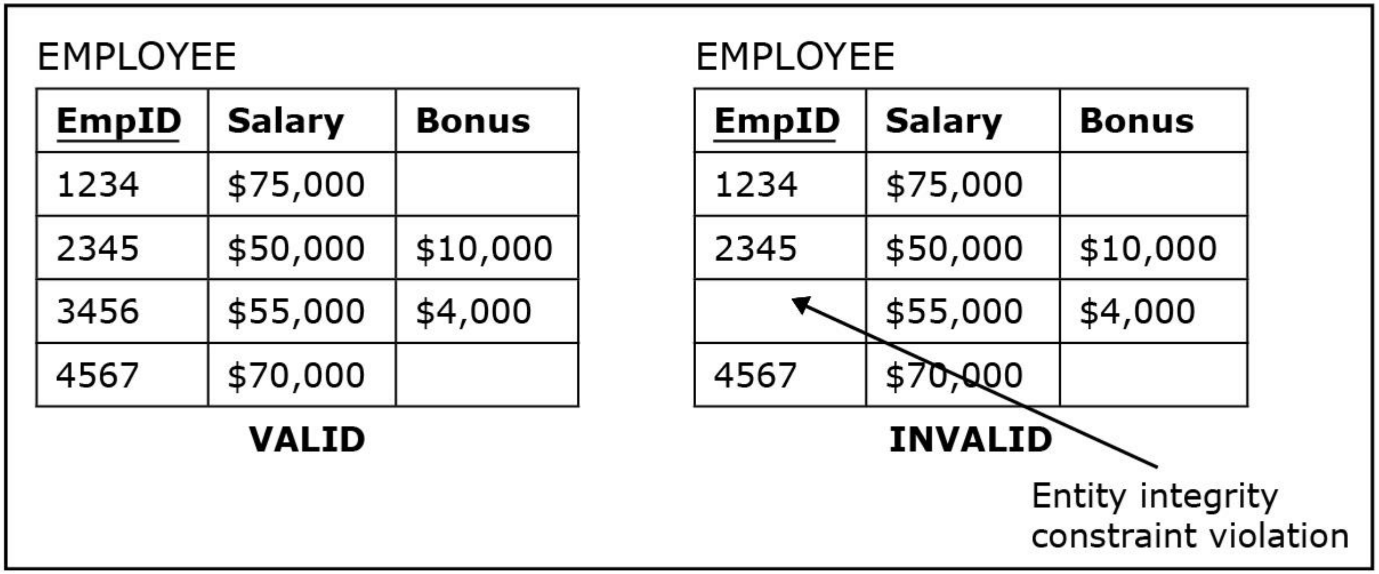


EMPLOYEE

EmpID	**Salary**	**Bonus**
1234	$75,000	
2345	$50,000	$10,000
3456	$55,000	$4,000
4567	$70,000	

ENTITY INTEGRITY CONSTRAINT

- **Entity integrity constraint** - *in a relational table, no primary key column can have null (empty) values*
  - A rule stating that **no primary key column can be optional**
  - Every RBMS enforces this rule



EMPLOYEE

EmpID	Salary	Bonus
1234	$75,000	
2345	$50,000	$10,000
3456	$55,000	$4,000
4567	$70,000	

VALID

EMPLOYEE

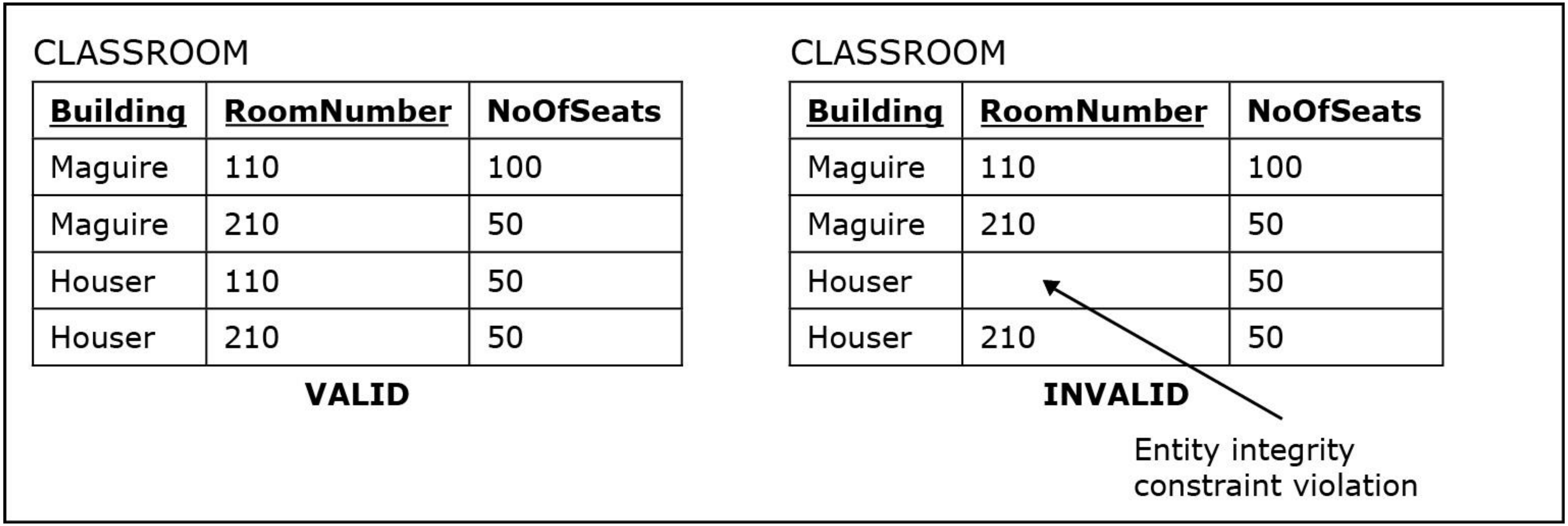
EmpID	Salary	Bonus
1234	$75,000	
2345	$50,000	$10,000
	$55,000	$4,000
4567	$70,000	

INVALID

Entity integrity constraint violation

ENTITY INTEGRITY CONSTRAINT

Entity integrity constraint — another compliance and violation example



CLASSROOM

Building	**RoomNumber**	**NoOfSeats**
Maguire	110	100
Maguire	210	50
Houser	110	50
Houser	210	50

VALID

CLASSROOM

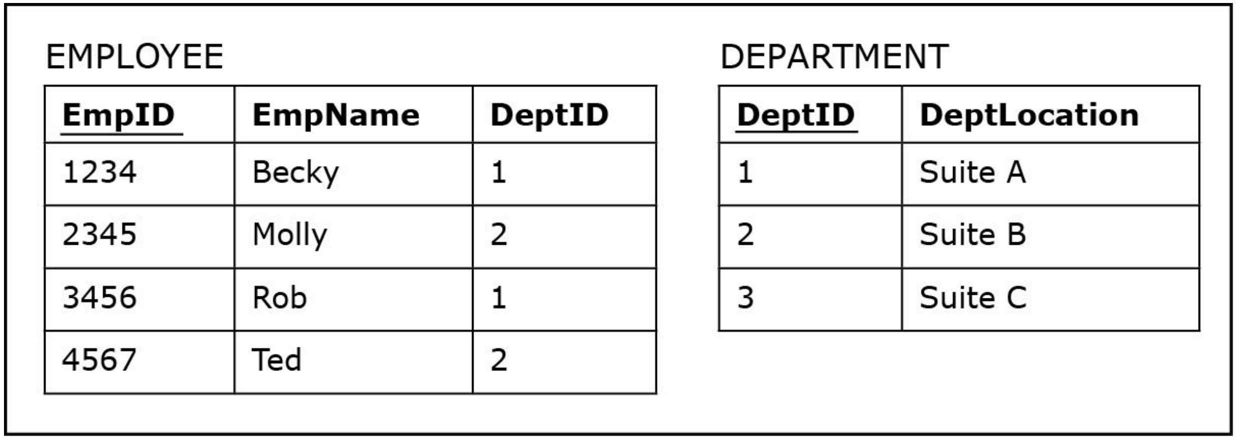
Building	**RoomNumber**	**NoOfSeats**
Maguire	110	100
Maguire	210	50
Houser		50
Houser	210	50

INVALID

Entity integrity constraint violation

FOREIGN KEY

- **Foreign key** - *column in a relation that refers to a primary key column in another (referred) relation*
  - A mechanism that is used to **depict relationships** in the relational database model
  - For every occurrence of a foreign key, the relational schema contains a line pointing *from* the *foreign key* to the *corresponding primary key*



EMPLOYEE

EmpID	**EmpName**	**DeptID**
1234	Becky	1
2345	Molly	2
3456	Rob	1
4567	Ted	2

DEPARTMENT

DeptID	**DeptLocation**
1	Suite A
2	Suite B
3	Suite C

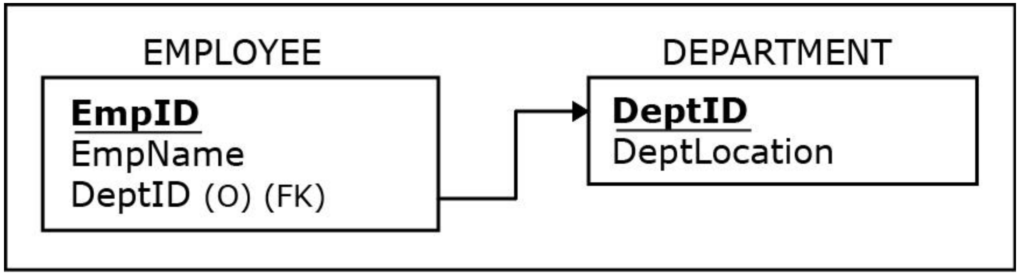
REFERENTIAL INTEGRITY CONSTRAINT

- **Referential integrity constraint** - *In each row of a relation containing a foreign key, the value of the* ***foreign key EITHER matches*** *one of the values in the* ***primary key*** *column of the referred relation* ***OR*** *the value of* ***the foreign key is null*** *(empty).*
  - Regulates the relationship between a table with a foreign key and a table with a primary key to which the foreign key refers

REFERENTIAL INTEGRITY CONSTRAINT

Referential integrity constraint compliance example

Two relations and a referential integrity constraint



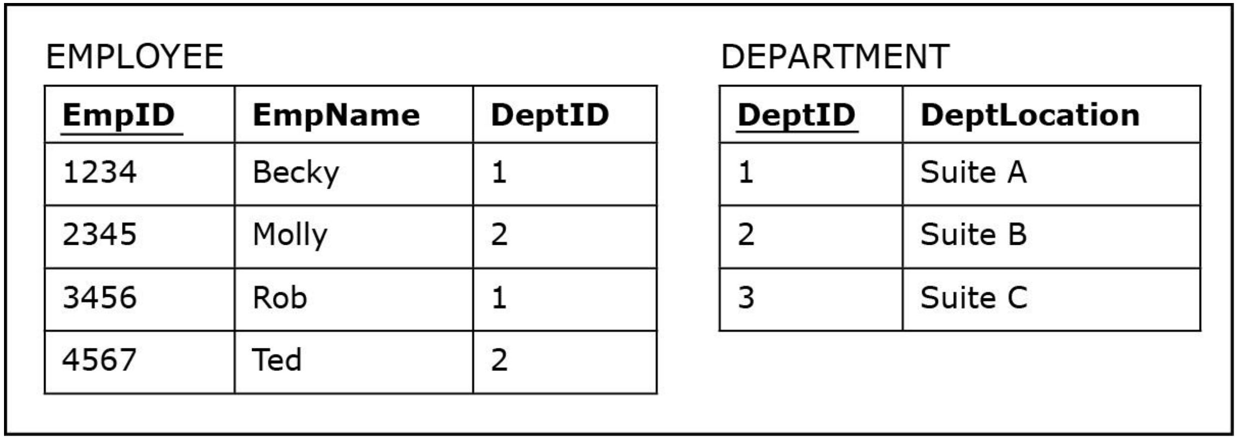
EMPLOYEE

EmpID
EmpName
DeptID (O) (FK)

DEPARTMENT

DeptID
DeptLocation

Data records in compliance with the referential integrity constraint



EMPLOYEE

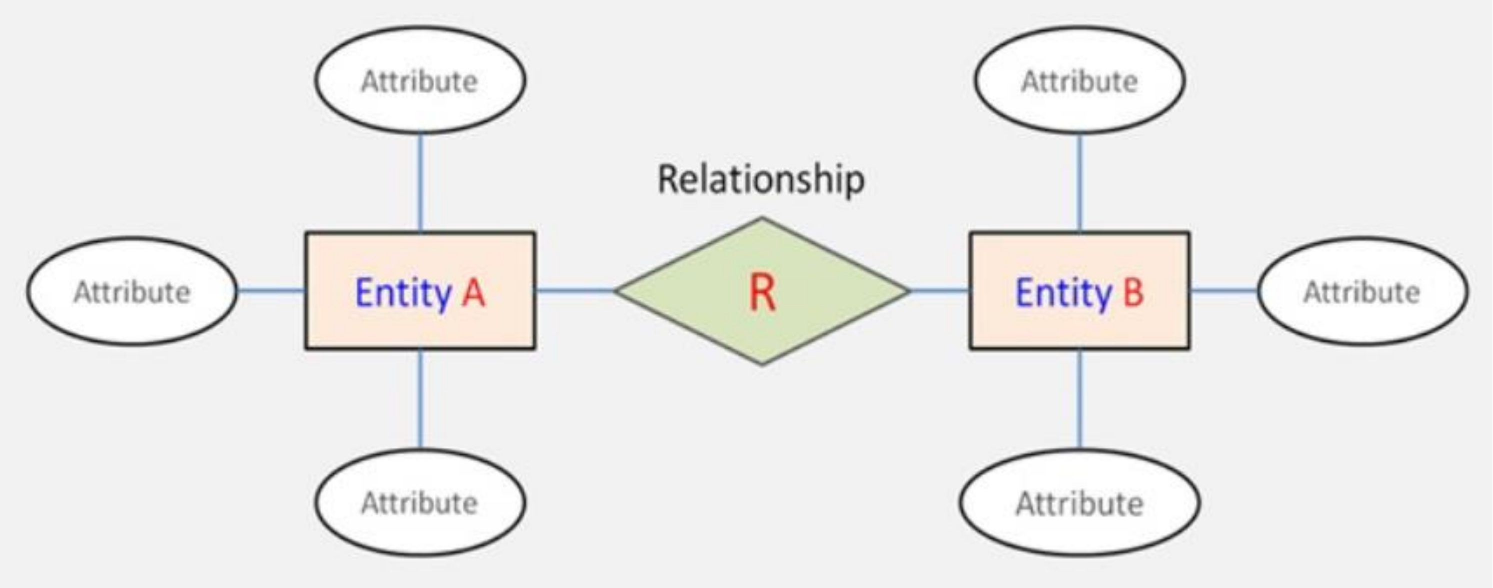
EmpID	**EmpName**	**DeptID**
1234	Becky	1
2345	Molly	2
3456	Rob	1
4567	Ted	2

DEPARTMENT

DeptID	**DeptLocation**
1	Suite A
2	Suite B
3	Suite C

ENTITY RELATIONSHIP DIAGRAM

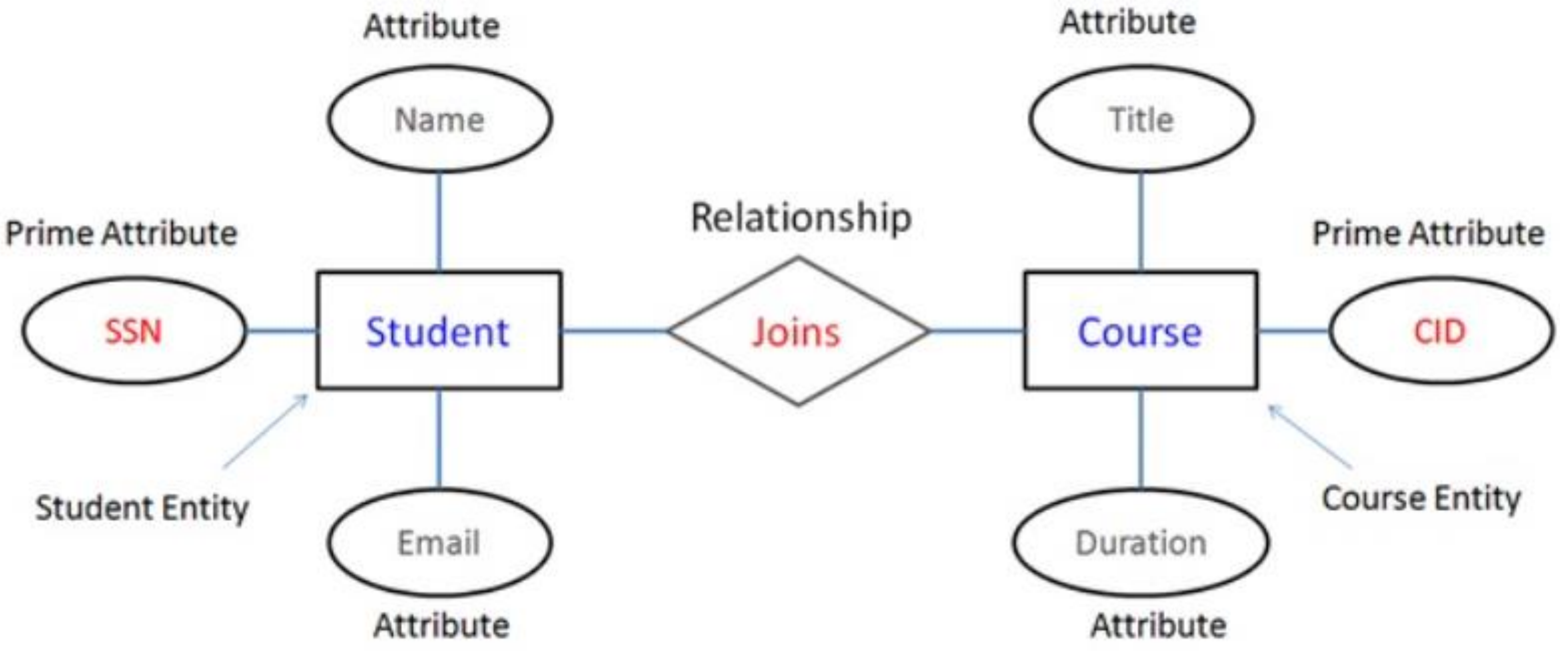
ERD: A type of graphical representation/flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system.



Source: https://www.learncomputerscienceonline.com/entity-relationship-diagram/

Intro to Database and ERD video

ENTITY RELATIONSHIP DIAGRAM

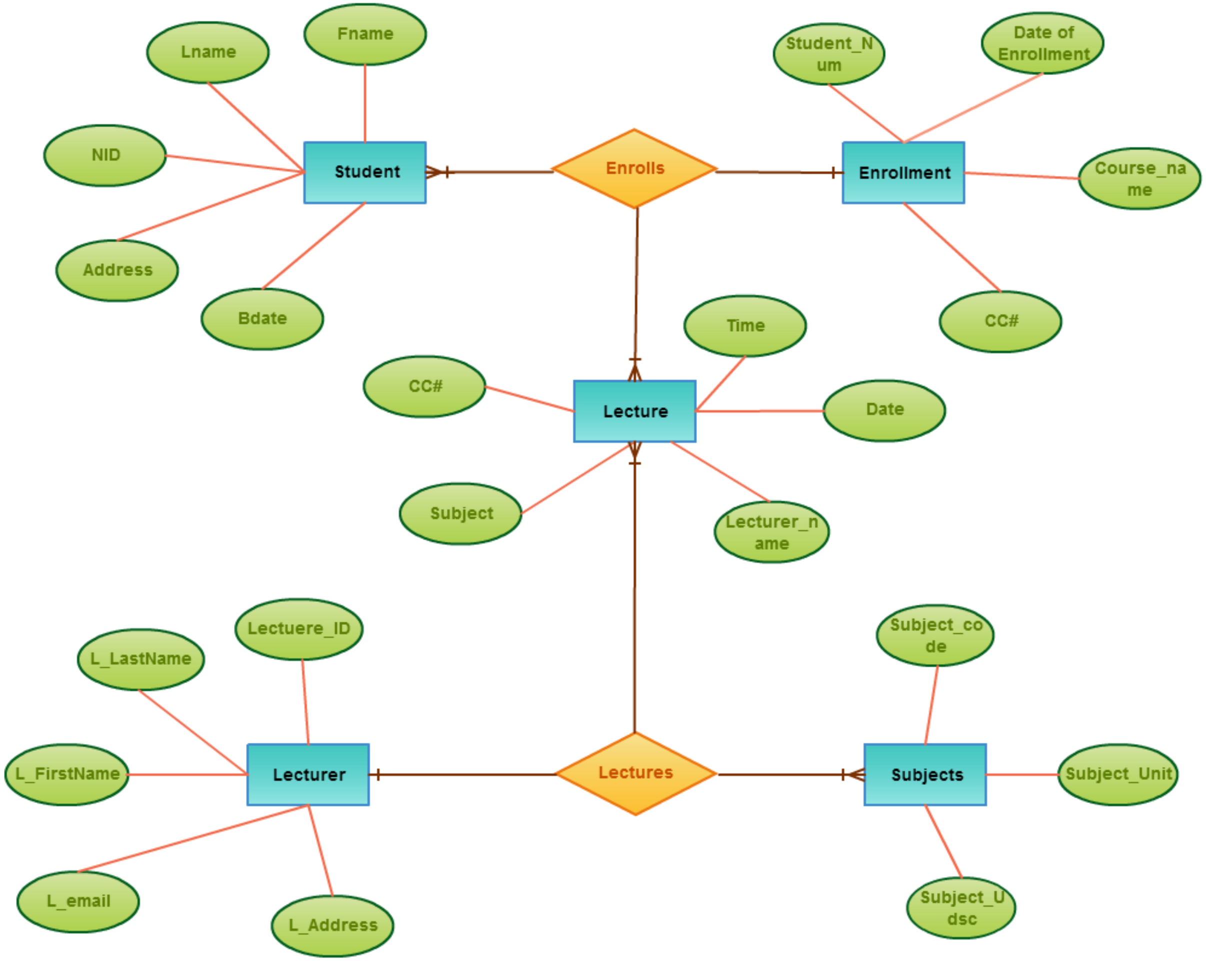


Source: https://www.learncomputerscienceonline.com/entity-relationship-diagram/

[Intro to Database and ERD video]

ENTITY RELATIONSHIP DIAGRAM

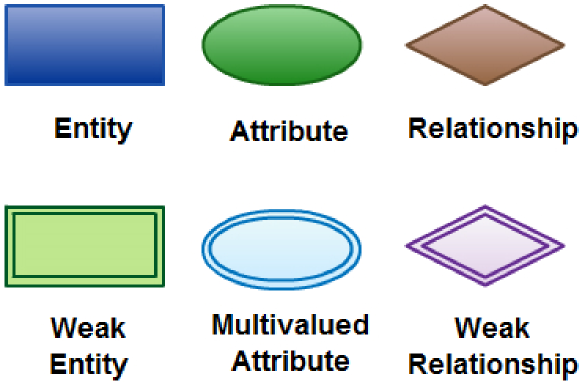
ER DIAGRAM FOR STUDENT ENROLLMENT SYSTEM



Source: https://creately.com/blog/diagrams/er-diagrams-tutorial/

ENTITY RELATIONSHIP DIAGRAM (ERD)

- A type of graphical representation/flowchart that illustrates how "entities" such as people, objects or concepts relate to each other within a system.
- ERD Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research.
- Use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes.
- They mirror grammatical structure, with entities as nouns and relationships as verbs.



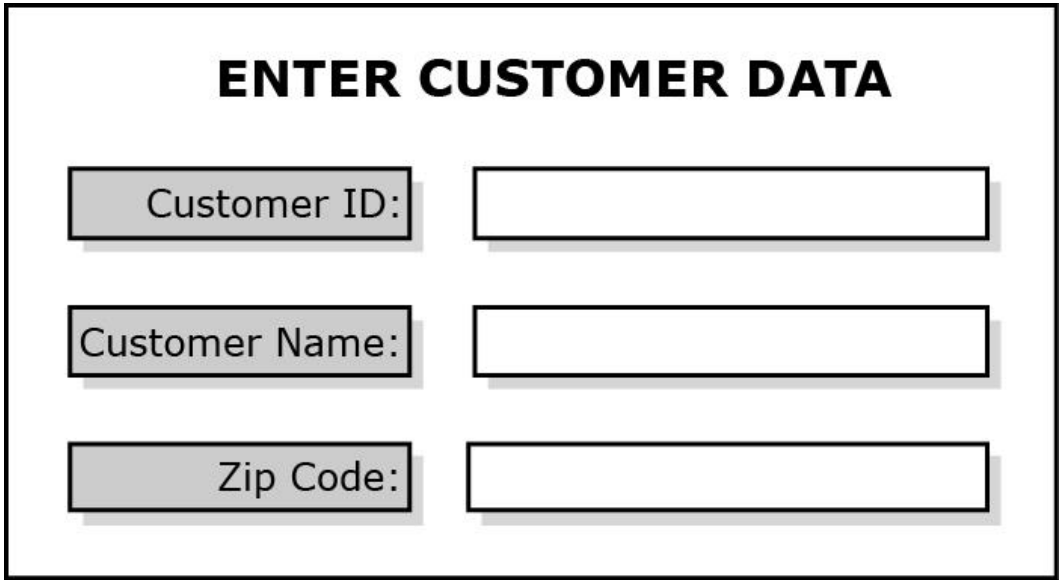
Source: https://www.lucidchart.com/pages/er-diagrams

DATABASE FRONT-END

- Database front-end
  - Provides access to the database for indirect use

- Form
  - Enables data input and retrieval for end users
  - Provides an interface into a database relation or query

Form – Example 1 (eg with SQL code)



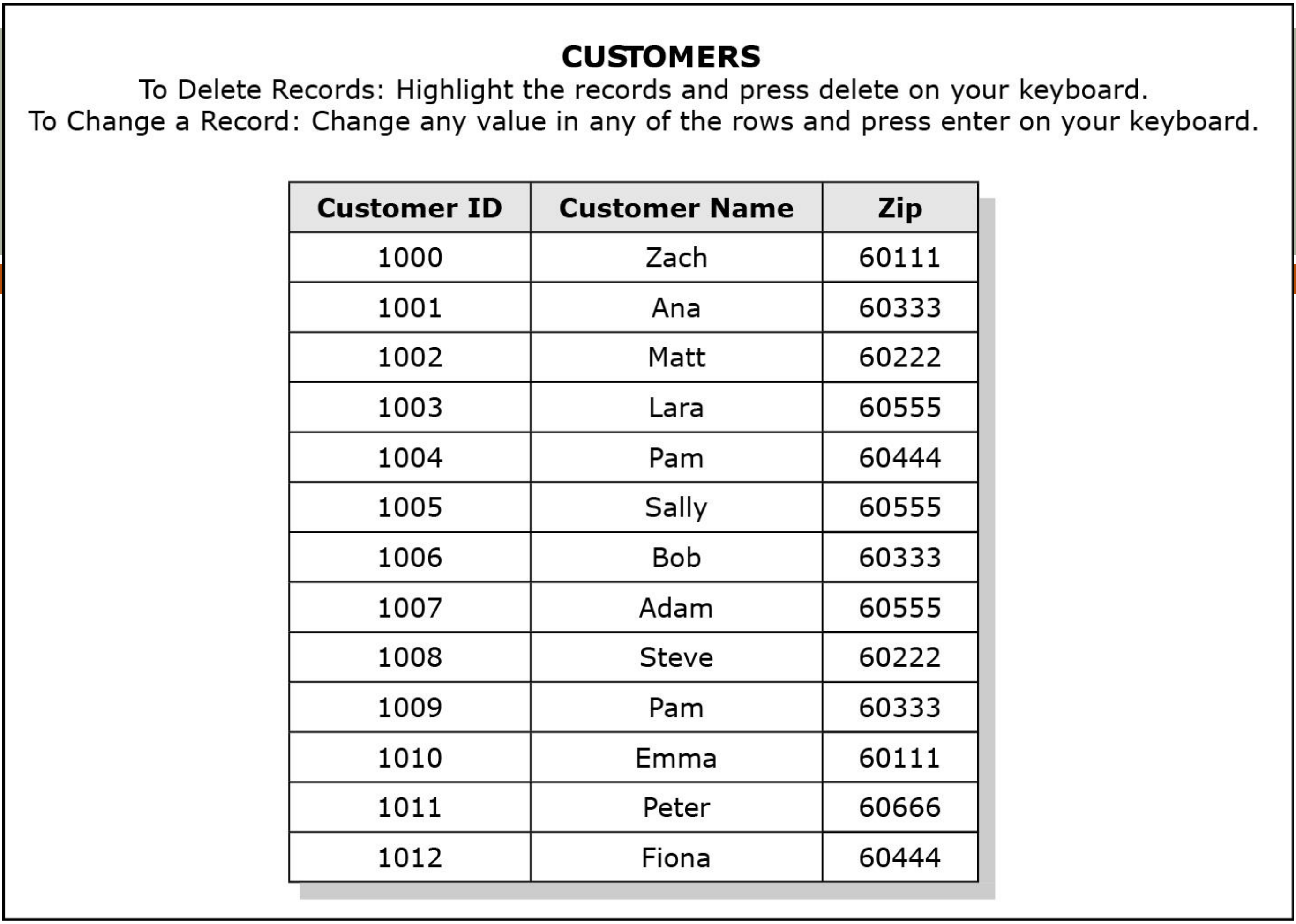
ENTER CUSTOMER DATA

Customer ID:	
Customer Name:	
Zip Code:	

SQL code generated:

```
INSERT INTO customer VALUES (…)
```

Form – Example 2



CUSTOMERS

To Delete Records: Highlight the records and press delete on your keyboard.
To Change a Record: Change any value in any of the rows and press enter on your keyboard.

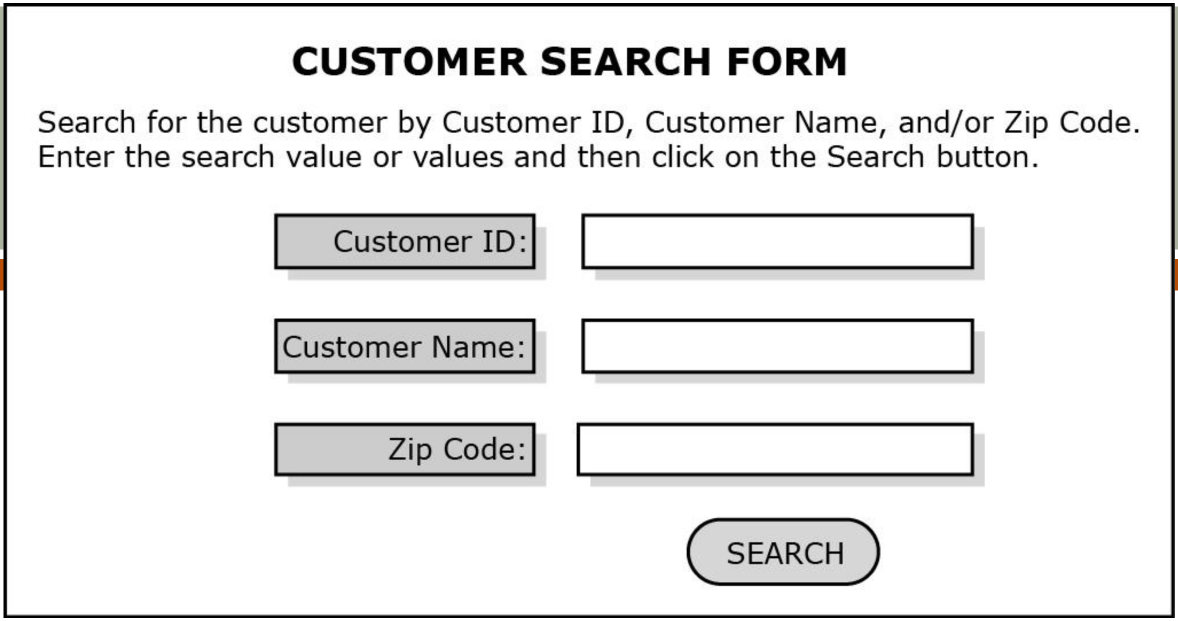
Customer ID	Customer Name	Zip
1000	Zach	60111
1001	Ana	60333
1002	Matt	60222
1003	Lara	60555
1004	Pam	60444
1005	Sally	60555
1006	Bob	60333
1007	Adam	60555
1008	Steve	60222
1009	Pam	60333
1010	Emma	60111
1011	Peter	60666
1012	Fiona	60444

Sample SQL code generated:

```
DELETE FROM customer
WHERE custid = '1012';

UPDATE customer
SET custname = 'Zachary'
WHERE custid = '1000';
```

Form – Example 3



CUSTOMER SEARCH FORM

Search for the customer by Customer ID, Customer Name, and/or Zip Code. Enter the search value or values and then click on the Search button.

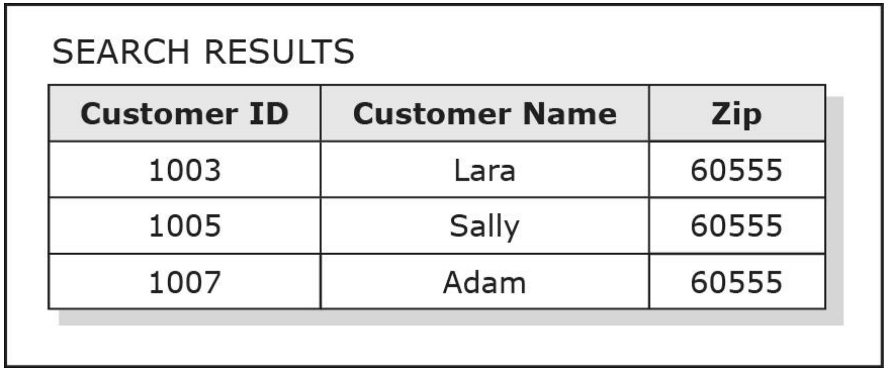
Customer ID:

Customer Name:

Zip Code:

Sample SQL code generated:

```
SELECT *
FROM customer
WHERE zip = '60555';
```

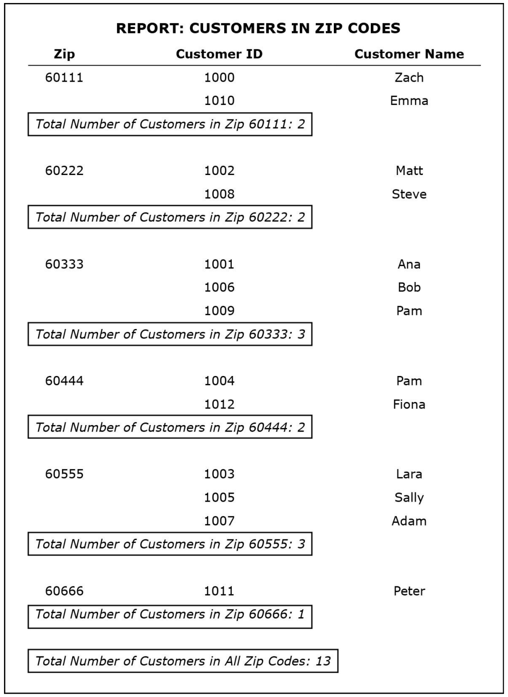


SEARCH RESULTS

Customer ID	Customer Name	Zip
1003	Lara	60555
1005	Sally	60555
1007	Adam	60555

DATABASE FRONT-END

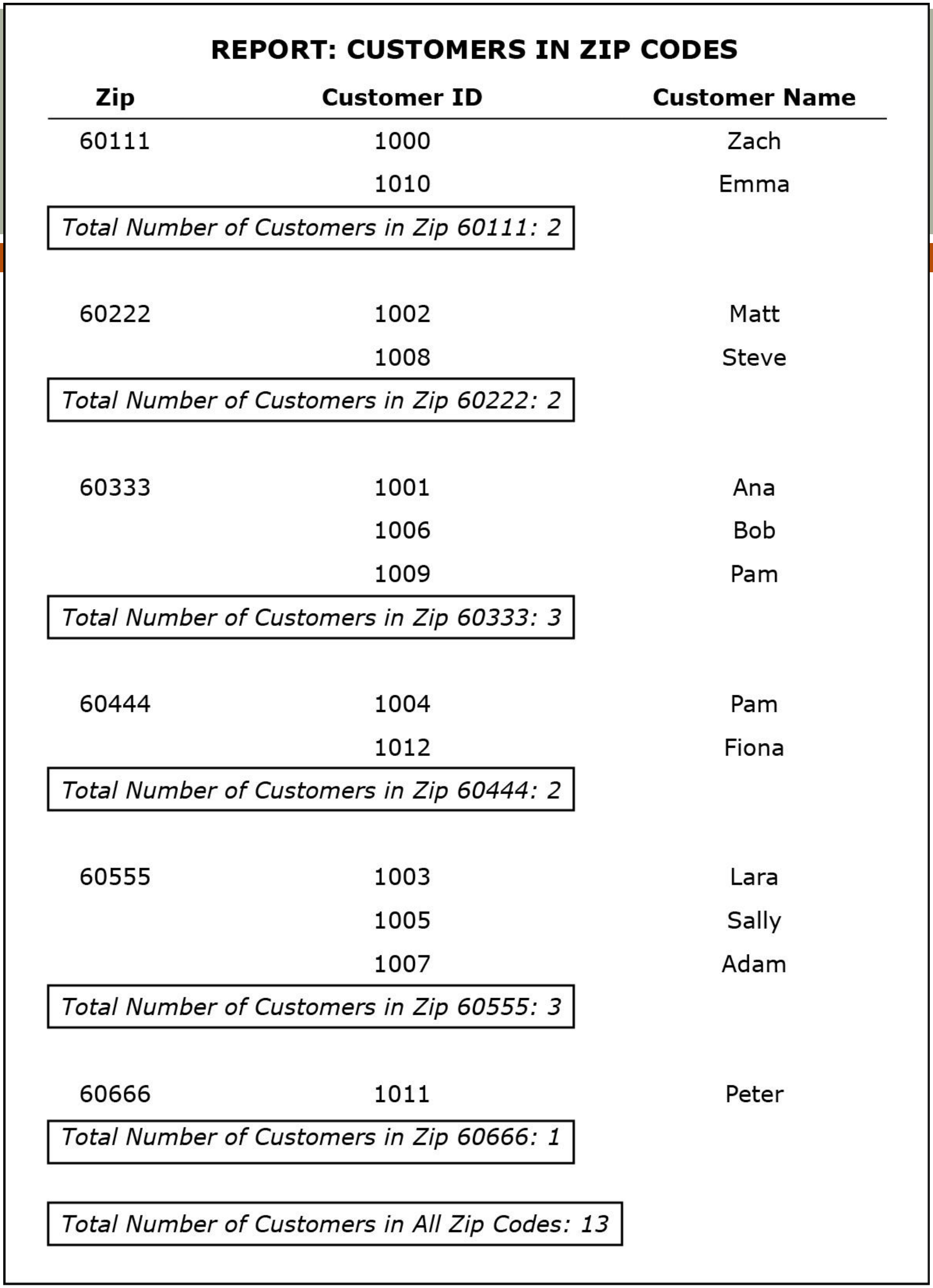
- Report
  - Presents the data and calculations on the data from one or more tables from the database in a formatted way
  - The data that is retrieved via reports is formatted and arranged to be displayed on the screen or printed as a hard copy



REPORT: CUSTOMERS IN ZIP CODES

Zip	Customer ID	Customer Name
60111	1000	Zach
	1010	Emma
Total Number of Customers in Zip 60111: 2		
60222	1002	Matt
	1008	Steve
Total Number of Customers in Zip 60222: 2		
60333	1001	Ana
	1006	Bob
	1009	Pam
Total Number of Customers in Zip 60333: 3		
60444	1004	Pam
	1012	Fiona
Total Number of Customers in Zip 60444: 2		
60555	1003	Lara
	1005	Sally
	1007	Adam
Total Number of Customers in Zip 60555: 3		
60666	1011	Peter
Total Number of Customers in Zip 60666: 1		
Total Number of Customers in All Zip Codes: 13		

Report – Example



REPORT: CUSTOMERS IN ZIP CODES

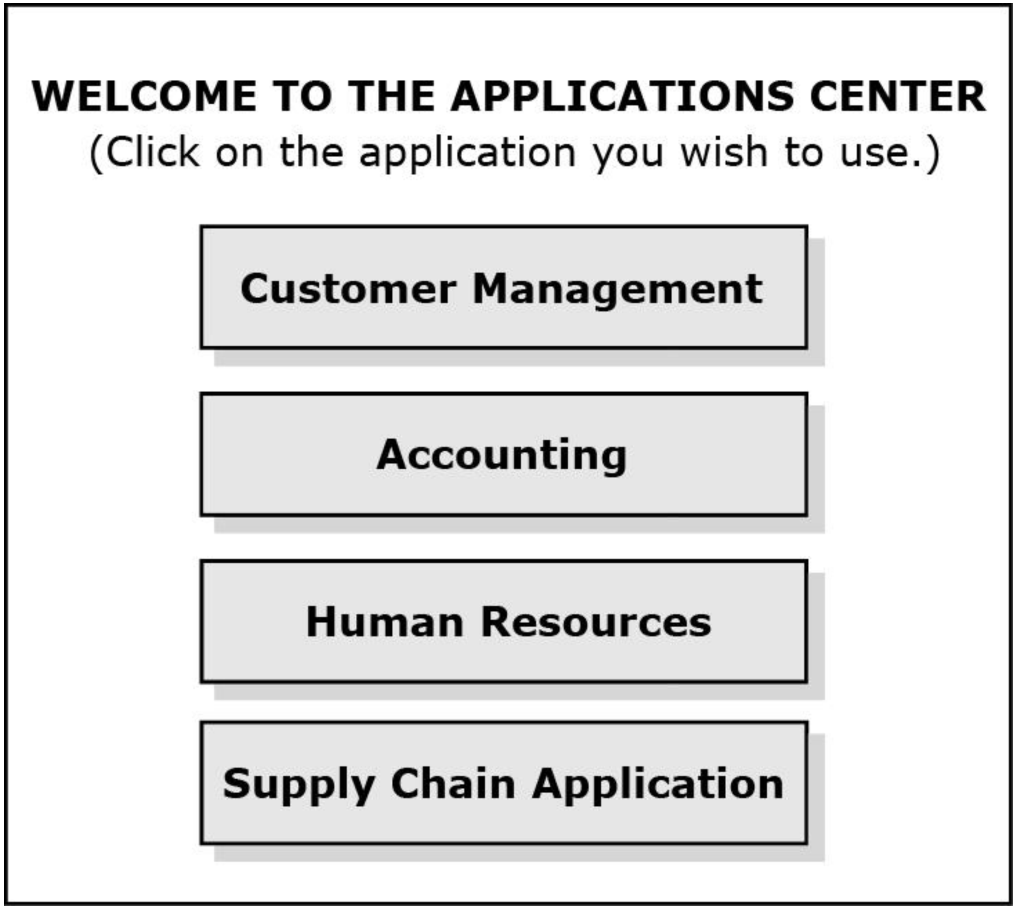
Zip	Customer ID	Customer Name
60111	1000	Zach
	1010	Emma
Total Number of Customers in Zip 60111: 2		
60222	1002	Matt
	1008	Steve
Total Number of Customers in Zip 60222: 2		
60333	1001	Ana
	1006	Bob
	1009	Pam
Total Number of Customers in Zip 60333: 3		
60444	1004	Pam
	1012	Fiona
Total Number of Customers in Zip 60444: 2		
60555	1003	Lara
	1005	Sally
	1007	Adam
Total Number of Customers in Zip 60555: 3		
60666	1011	Peter
Total Number of Customers in Zip 60666: 1		
Total Number of Customers in All Zip Codes: 13		

DATABASE FRONT-END

- In addition to the forms and reports, database front-end applications can include many other components and functionalities, such as:
  - menus
  - charts
  - graphs
  - maps
  - etc.
- The choice of how many different components to use and to what extent is driven by the needs of the end-users
- A database can have multiple sets of front-end applications for different purposes or groups of end-users

Front-end interface – Example

An example of an interface to a collection of database front-end applications

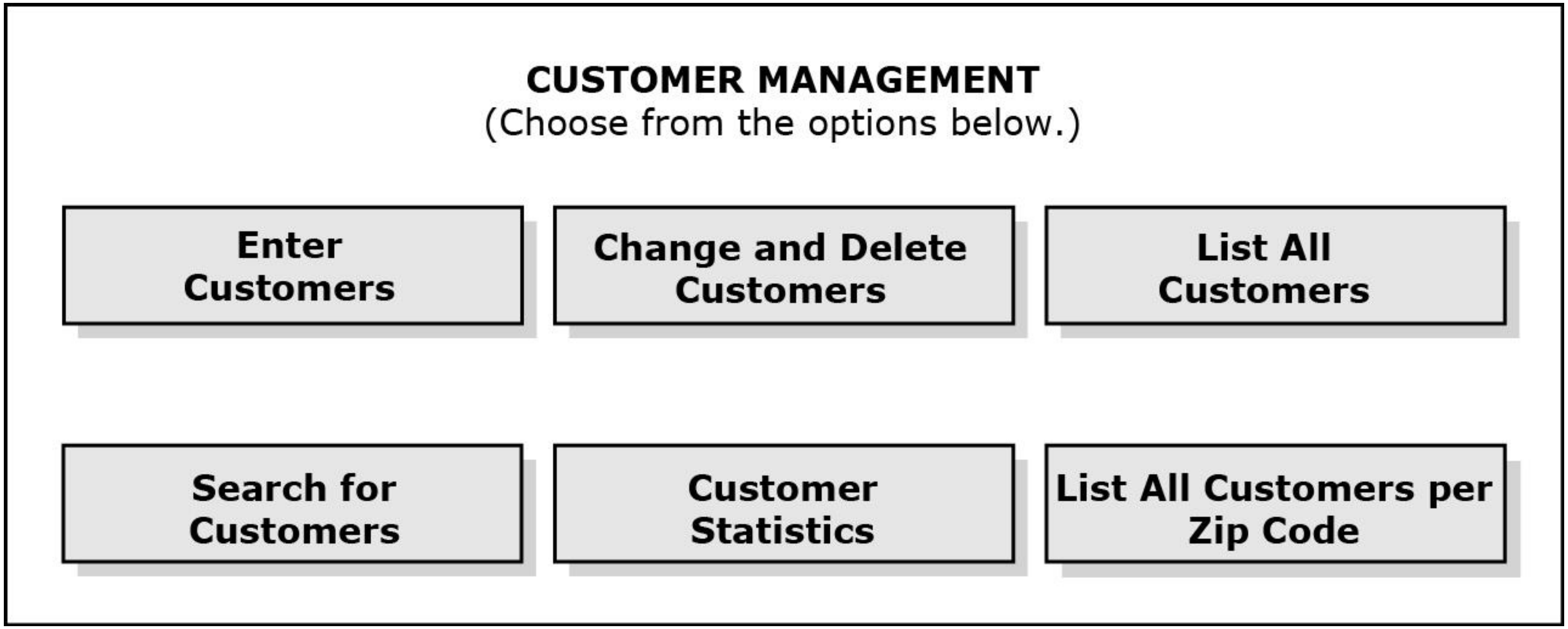


WELCOME TO THE APPLICATIONS CENTER

(Click on the application you wish to use.)

Front-end interface – Example

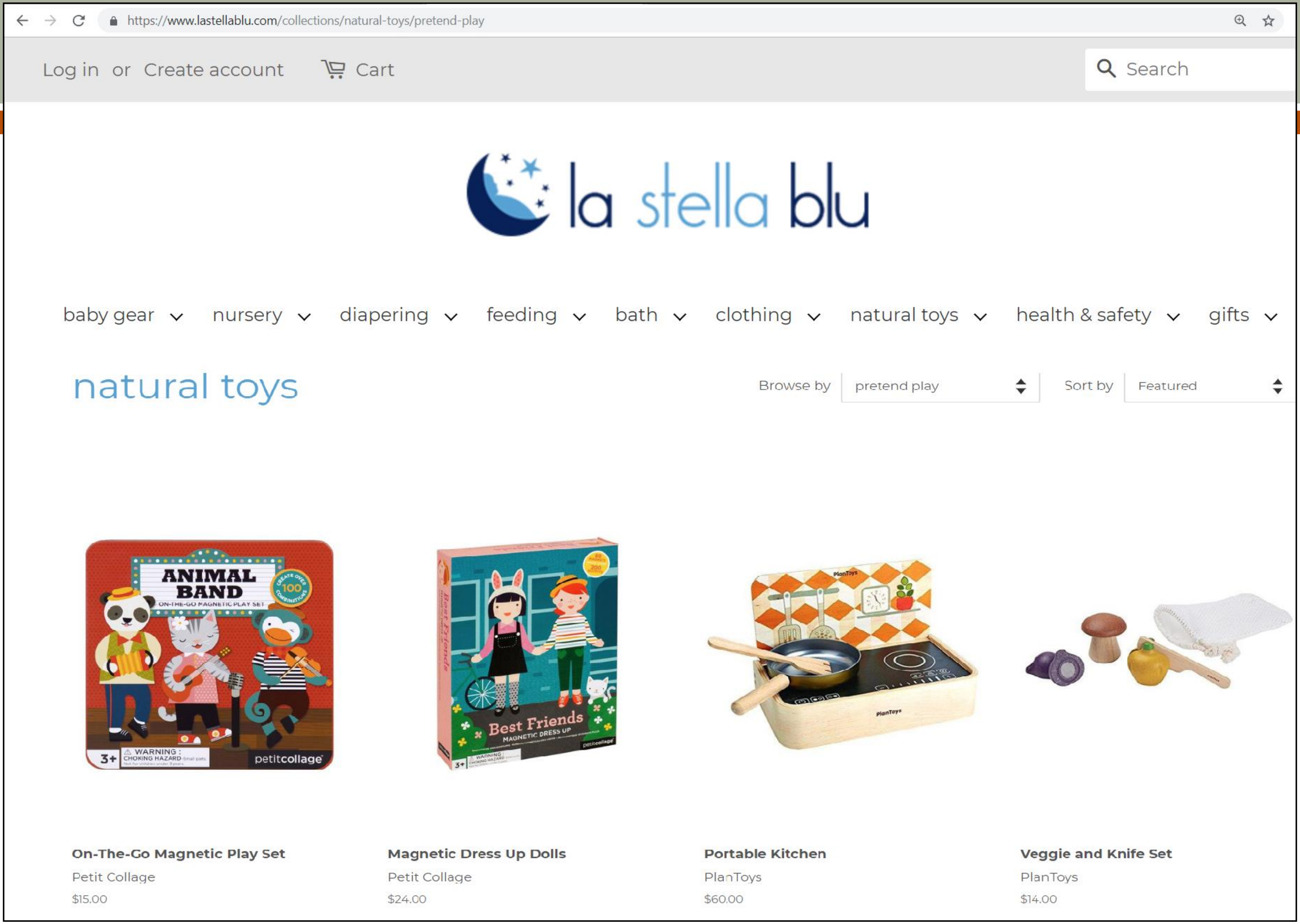
An example of an interface to a database front-end application



CUSTOMER MANAGEMENT

(Choose from the options below.)

Web page - Example



natural toys

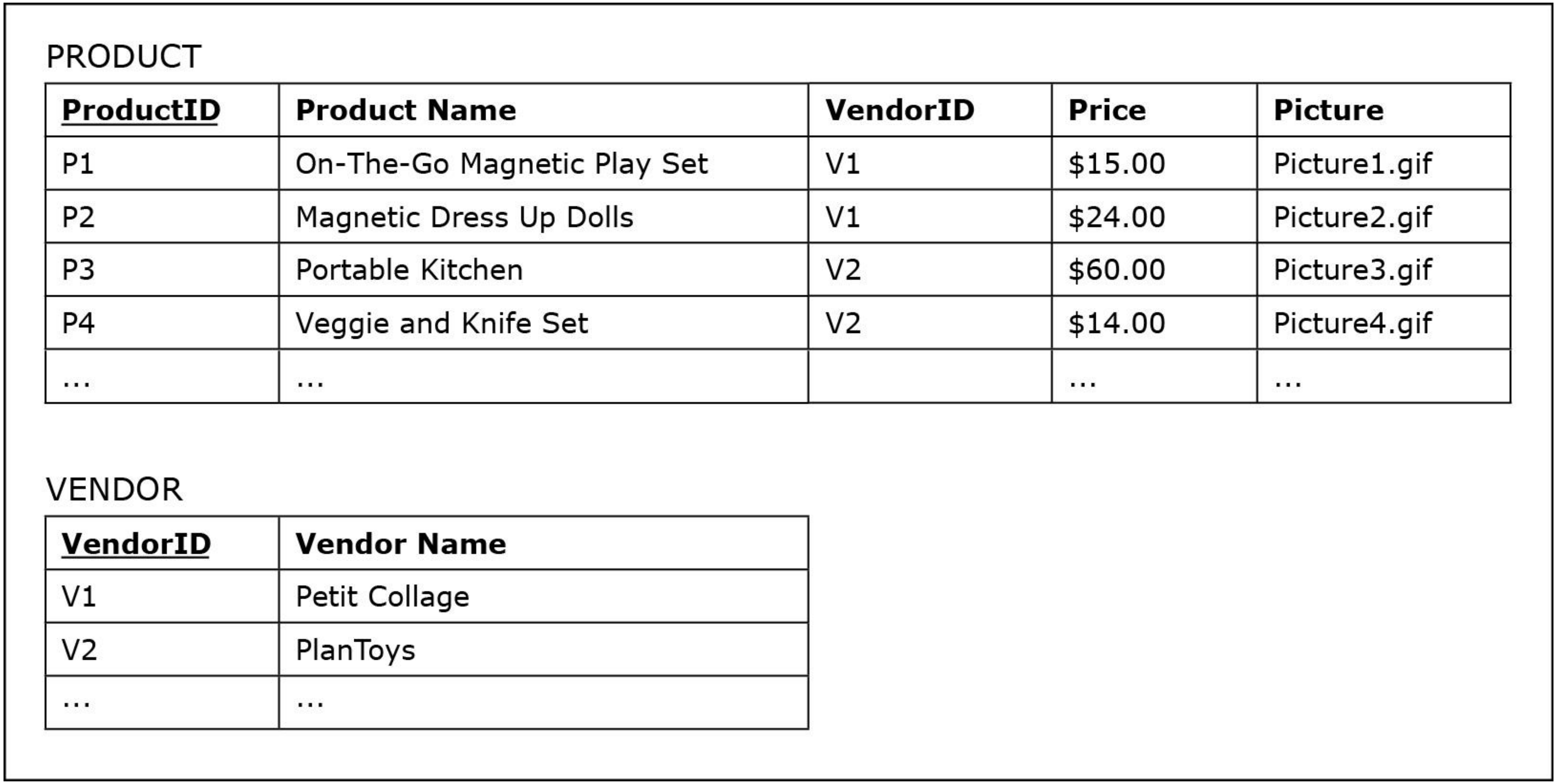
On-The-Go Magnetic Play Set
Petit Collage
$15.00

Magnetic Dress Up Dolls
Petit Collage
$24.00

Portable Kitchen
PlanToys
$60.00

Veggie and Knife Set
PlanToys
$14.00

Web page - Example
Database relations storing the content of the Web site



PRODUCT

ProductID	**Product Name**	**VendorID**	**Price**	**Picture**
P1	On-The-Go Magnetic Play Set	V1	$15.00	Picture1.gif
P2	Magnetic Dress Up Dolls	V1	$24.00	Picture2.gif
P3	Portable Kitchen	V2	$60.00	Picture3.gif
P4	Veggie and Knife Set	V2	$14.00	Picture4.gif
...	...		...	...

VENDOR

VendorID	**Vendor Name**
V1	Petit Collage
V2	PlanToys
...	...

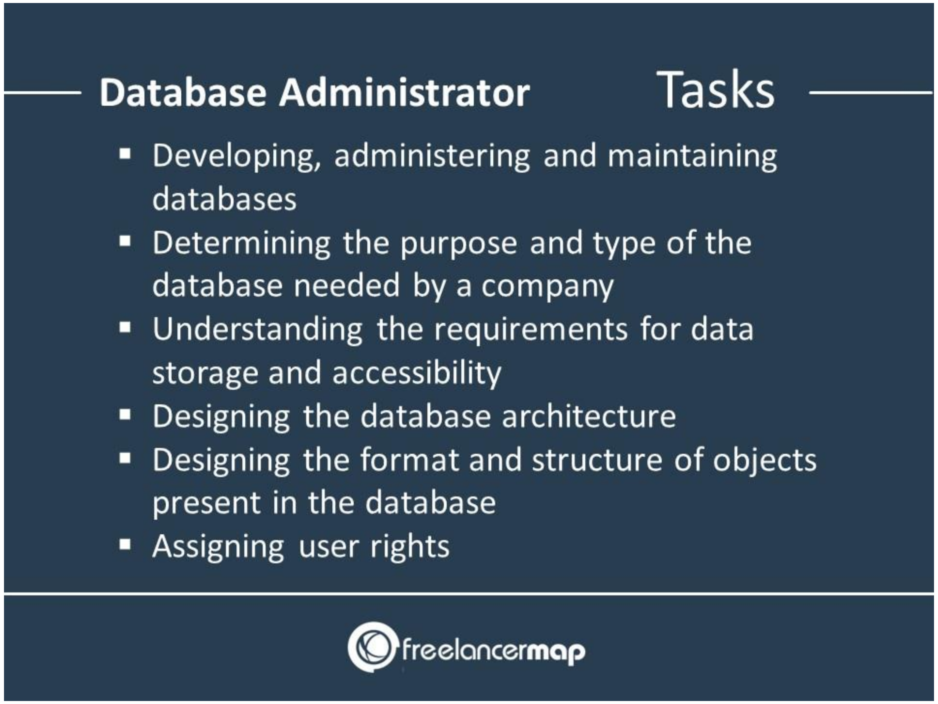
PEOPLE INVOLVED WITH DATABASE SYSTEMS

- Database analysts, designers, and developers
  - **Database analysts** - involved in the requirements collection, definition, and visualisation stage
  - **Database designers** (a.k.a. **database modelers** or **architects**) - involved in the database modeling stage
  - **Database developers** – in charge of implementing the database model as a functioning database using the DBMS software

- Front-end applications analysts and developers
  - **Front-end application analysts** - in charge of collecting and defining requirements for front-end applications
  - **Front-end applications developers** - in charge of creating the front-end applications

PEOPLE INVOLVED WITH DATABASE SYSTEMS

- **Database administrators (DBAs)** - perform the tasks related to the maintenance and administration of a database system
- **Database end users** - use a database system to support their work- or life-related tasks and processes



Database Administrator Tasks

- Developing, administering and maintaining databases
- Determining the purpose and type of the database needed by a company
- Understanding the requirements for data storage and accessibility
- Designing the database architecture
- Designing the format and structure of objects present in the database
- Assigning user rights

freelancermap

Source: https://www.freelancermap.com/blog/what-does-database-administrator-do/

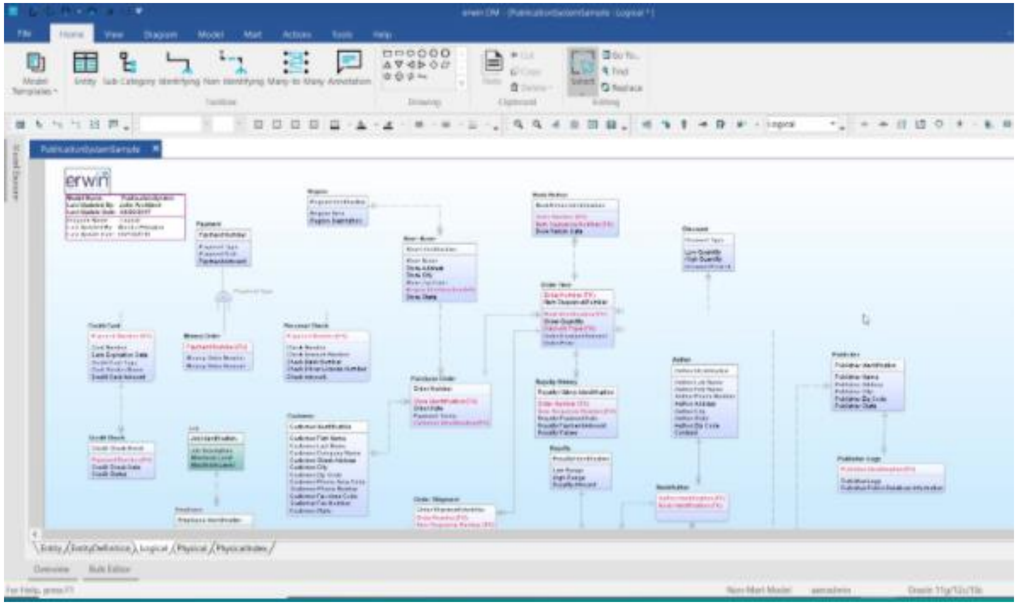
Data Modelling Tool: Erwin DM

- https://erwin.com/products/erwin-data-modeler/

  - Automated Data Model & Database Schema Generation

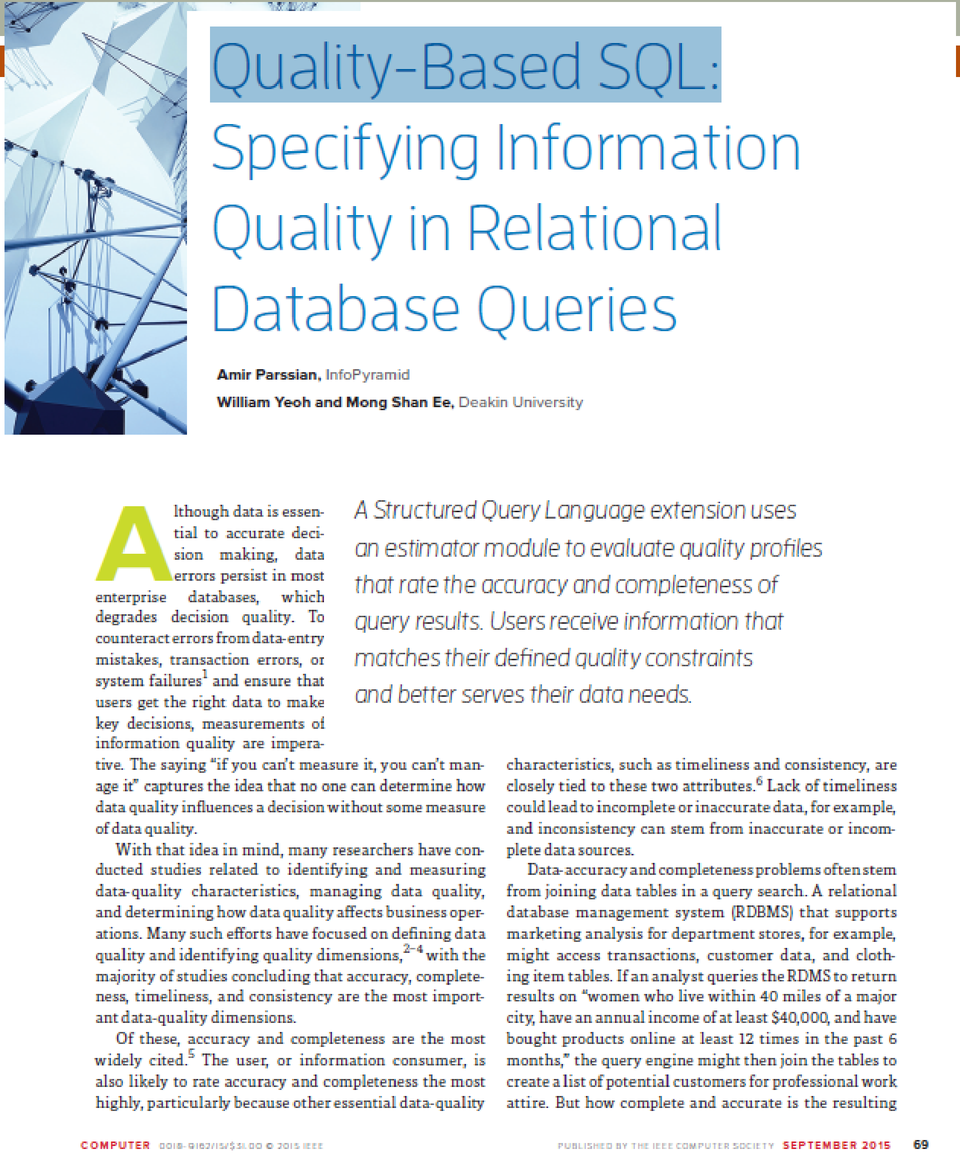
- Top 10 Features of Erwin Data Modeler 2020 R1:

https://www.youtube.com/watch?v=GwuSo59PqOU



Database Administrator
What my friends think I do
What my customers think I do
What my boss thinks I do
FARMVILLE
What my mom thinks I do
What I think I do
Oracle 11g
Oracle 11g DUMMIES
What I really do

Research Insight: Quality-Based SQL (QSQL)



Quality-Based SQL: Specifying Information Quality in Relational Database Queries

Amir Parssian, InfoPyramid

William Yeoh and Mong Shan Ee, Deakin University

A Structured Query Language extension uses an estimator module to evaluate quality profiles that rate the accuracy and completeness of query results. Users receive information that matches their defined quality constraints and better serves their data needs.

Although data is essential to accurate decision making, data errors persist in most enterprise databases, which degrades decision quality. To counteract errors from data-entry mistakes, transaction errors, or system failures[1] and ensure that users get the right data to make key decisions, measurements of information quality are imperative. The saying "if you can't measure it, you can't manage it" captures the idea that no one can determine how data quality influences a decision without some measure of data quality.

With that idea in mind, many researchers have conducted studies related to identifying and measuring data-quality characteristics, managing data quality, and determining how data quality affects business operations. Many such efforts have focused on defining data quality and identifying quality dimensions,[2–4] with the majority of studies concluding that accuracy, completeness, timeliness, and consistency are the most important data-quality dimensions.

Of these, accuracy and completeness are the most widely cited.[5] The user, or information consumer, is also likely to rate accuracy and completeness the most highly, particularly because other essential data-quality characteristics, such as timeliness and consistency, are closely tied to these two attributes.[6] Lack of timeliness could lead to incomplete or inaccurate data, for example, and inconsistency can stem from inaccurate or incomplete data sources.

Data-accuracy and completeness problems often stem from joining data tables in a query search. A relational database management system (RDBMS) that supports marketing analysis for department stores, for example, might access transactions, customer data, and clothing item tables. If an analyst queries the RDMS to return results on "women who live within 40 miles of a major city, have an annual income of at least $40,000, and have bought products online at least 12 times in the past 6 months," the query engine might then join the tables to create a list of potential customers for professional work attire. But how complete and accurate is the resulting

COMPUTER 0018-9162/15/$31.00 © 2015 IEEE PUBLISHED BY THE IEEE COMPUTER SOCIETY SEPTEMBER 2015 69

https://doi.org/10.1109/MC.2015.264