An Ancient Story

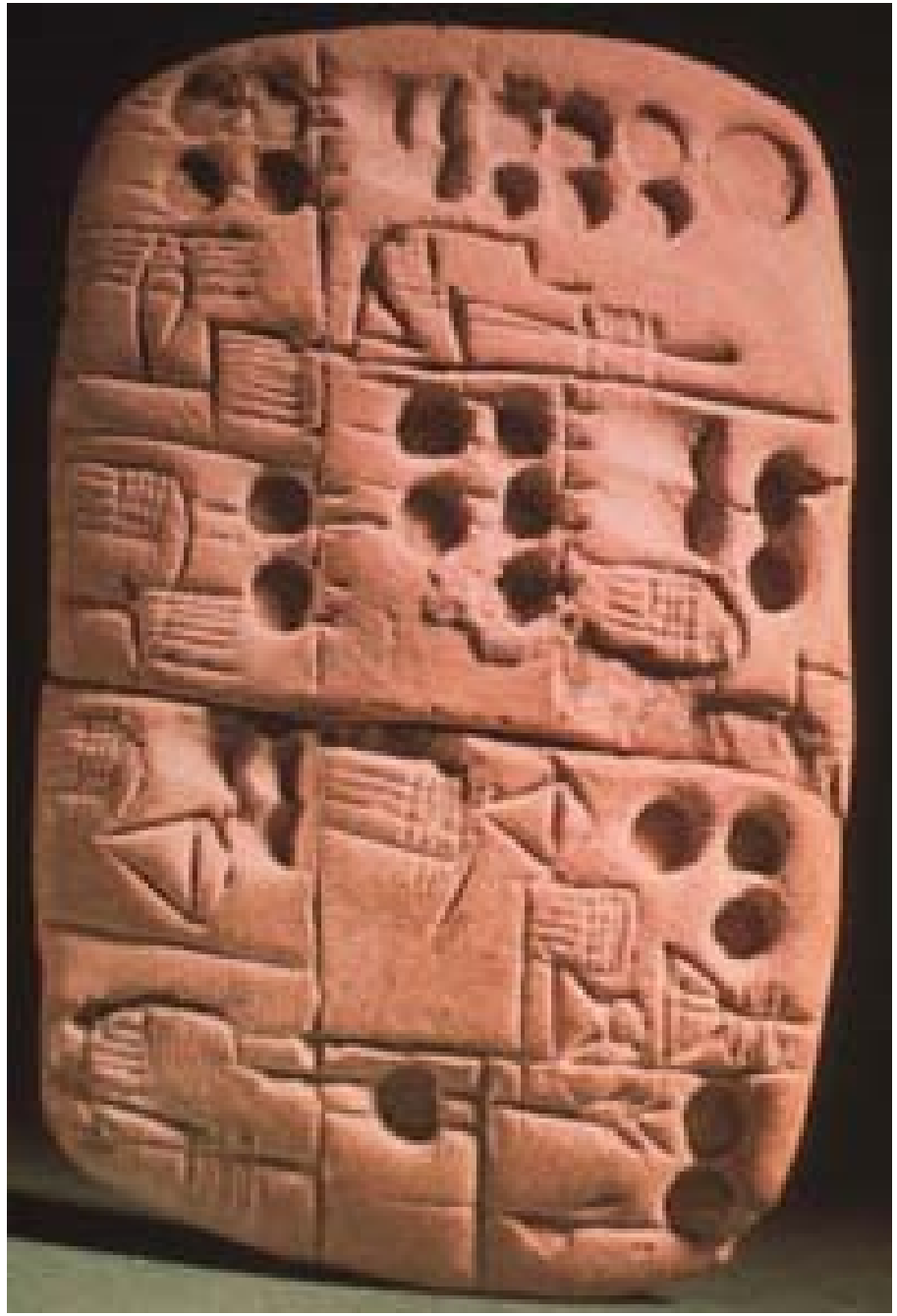
by Angel Marchev, Jr.

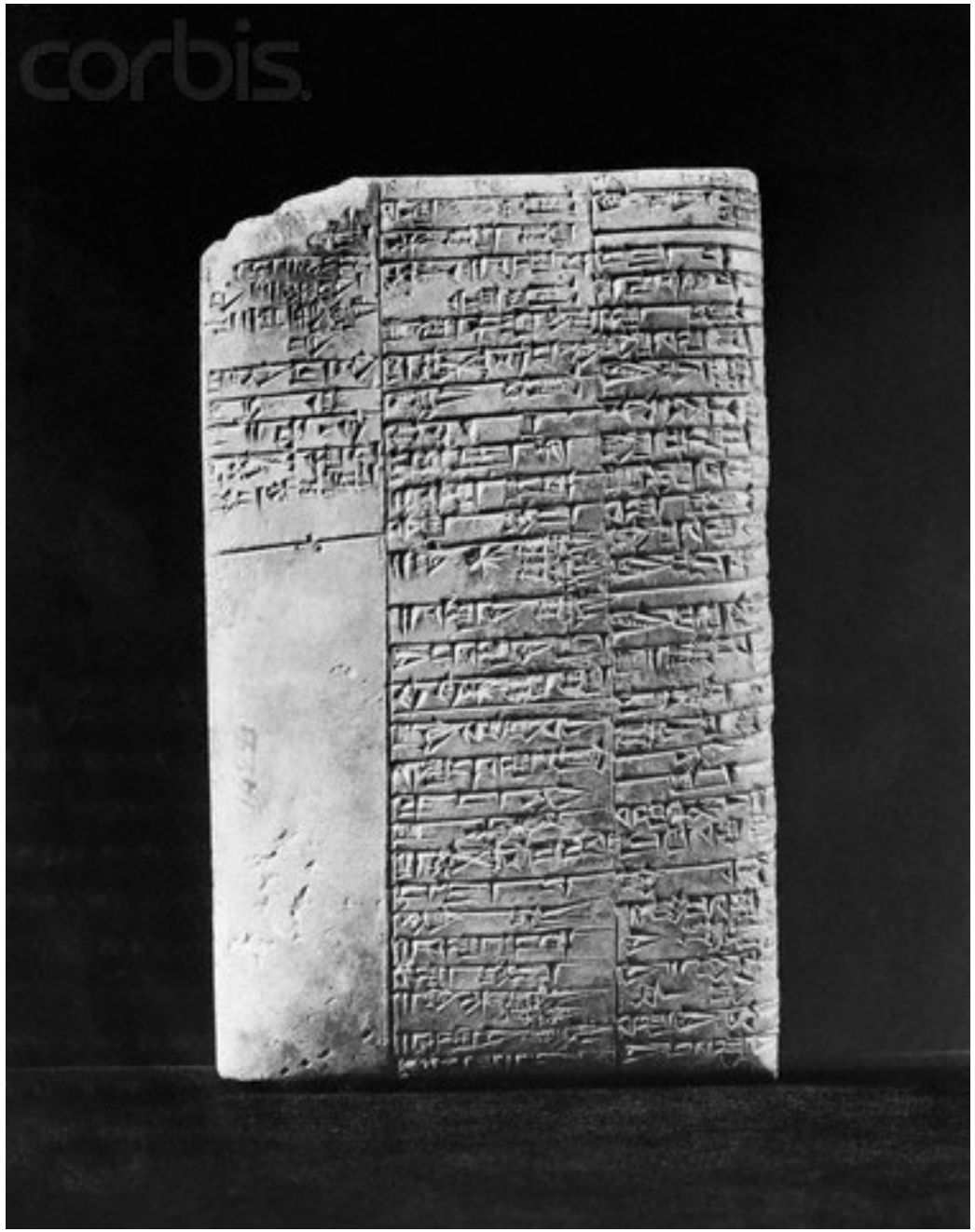
Babylon

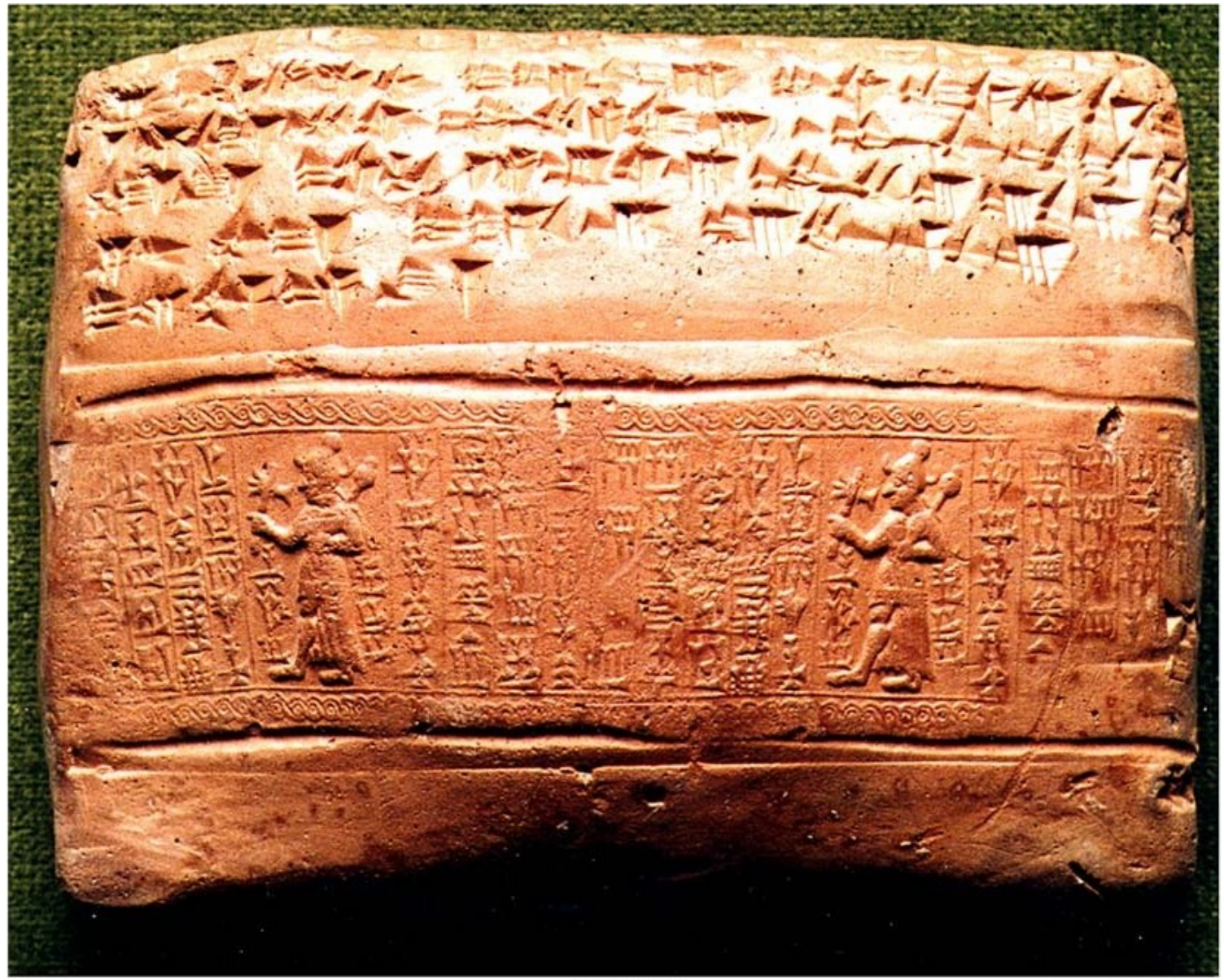
Ur

MS 4638
Bulla-envelope with 1 plain token inside.
Near East, ca. 3700-3200 BC

MS 4631
Bulla-envelope with 11 plain and complex tokens inside.
Near East, ca. 3700-3200 BC



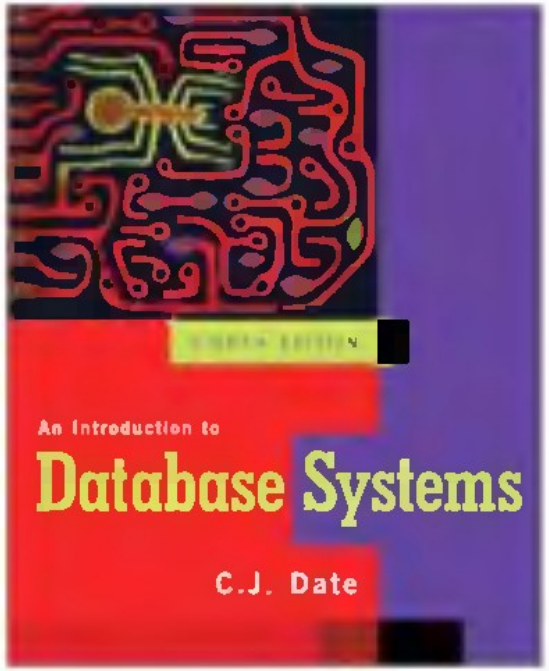

corbis



MS 1955/1
International judgement made before Initeshshub, King of Carchemish and Shaushgamuwa King of Amurru.
Rollseal depicting the deity Sharruma advancing left, holding a double axe and a sceptre.
Carchemish, Syria, 1250-1240 BC.

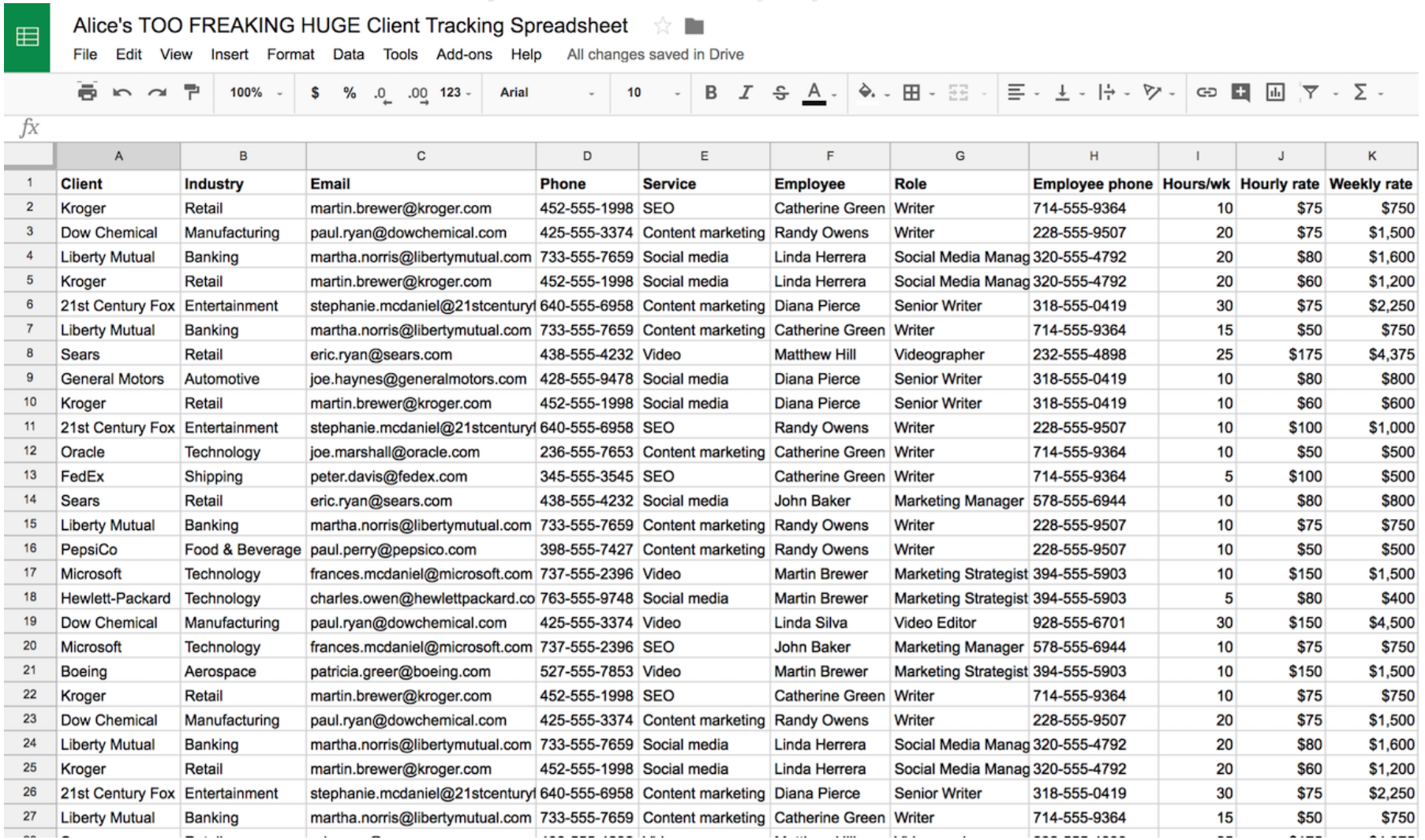
Database introduction

by Angel Marchev, Jr.



Excel 'tables'

The idea of a table shouldn't be new to you if you have used Excel, as that has rows and columns of information and the structure of a SQL Server table is similar to that of an Excel spreadsheet



Alice's TOO FREAKING HUGE Client Tracking Spreadsheet

File Edit View Insert Format Data Tools Add-ons Help All changes saved in Drive

	A	B	C	D	E	F	G	H	I	J	K
1	**Client**	**Industry**	**Email**	**Phone**	**Service**	**Employee**	**Role**	**Employee phone**	**Hours/wk**	**Hourly rate**	**Weekly rate**
2	Kroger	Retail	martin.brewer@kroger.com	452-555-1998	SEO	Catherine Green	Writer	714-555-9364	10	$75	$750
3	Dow Chemical	Manufacturing	paul.ryan@dowchemical.com	425-555-3374	Content marketing	Randy Owens	Writer	228-555-9507	20	$75	$1,500
4	Liberty Mutual	Banking	martha.norris@libertymutual.com	733-555-7659	Social media	Linda Herrera	Social Media Manag	320-555-4792	20	$80	$1,600
5	Kroger	Retail	martin.brewer@kroger.com	452-555-1998	Social media	Linda Herrera	Social Media Manag	320-555-4792	20	$60	$1,200
6	21st Century Fox	Entertainment	stephanie.mcdaniel@21stcentury	640-555-6958	Content marketing	Diana Pierce	Senior Writer	318-555-0419	30	$75	$2,250
7	Liberty Mutual	Banking	martha.norris@libertymutual.com	733-555-7659	Content marketing	Catherine Green	Writer	714-555-9364	15	$50	$750
8	Sears	Retail	eric.ryan@sears.com	438-555-4232	Video	Matthew Hill	Videographer	232-555-4898	25	$175	$4,375
9	General Motors	Automotive	joe.haynes@generalmotors.com	428-555-9478	Social media	Diana Pierce	Senior Writer	318-555-0419	10	$80	$800
10	Kroger	Retail	martin.brewer@kroger.com	452-555-1998	Social media	Diana Pierce	Senior Writer	318-555-0419	10	$60	$600
11	21st Century Fox	Entertainment	stephanie.mcdaniel@21stcentury	640-555-6958	SEO	Randy Owens	Writer	228-555-9507	10	$100	$1,000
12	Oracle	Technology	joe.marshall@oracle.com	236-555-7653	Content marketing	Catherine Green	Writer	714-555-9364	10	$50	$500
13	FedEx	Shipping	peter.davis@fedex.com	345-555-3545	SEO	Catherine Green	Writer	714-555-9364	5	$100	$500
14	Sears	Retail	eric.ryan@sears.com	438-555-4232	Social media	John Baker	Marketing Manager	578-555-6944	10	$80	$800
15	Liberty Mutual	Banking	martha.norris@libertymutual.com	733-555-7659	Content marketing	Randy Owens	Writer	228-555-9507	10	$75	$750
16	PepsiCo	Food & Beverage	paul.perry@pepsico.com	398-555-7427	Content marketing	Randy Owens	Writer	228-555-9507	10	$50	$500
17	Microsoft	Technology	frances.mcdaniel@microsoft.com	737-555-2396	Video	Martin Brewer	Marketing Strategist	394-555-5903	10	$150	$1,500
18	Hewlett-Packard	Technology	charles.owen@hewlettpackard.co	763-555-9748	Social media	Martin Brewer	Marketing Strategist	394-555-5903	5	$80	$400
19	Dow Chemical	Manufacturing	paul.ryan@dowchemical.com	425-555-3374	Video	Linda Silva	Video Editor	928-555-6701	30	$150	$4,500
20	Microsoft	Technology	frances.mcdaniel@microsoft.com	737-555-2396	SEO	John Baker	Marketing Manager	578-555-6944	10	$75	$750
21	Boeing	Aerospace	patricia.greer@boeing.com	527-555-7853	Video	Martin Brewer	Marketing Strategist	394-555-5903	10	$150	$1,500
22	Kroger	Retail	martin.brewer@kroger.com	452-555-1998	SEO	Catherine Green	Writer	714-555-9364	10	$75	$750
23	Dow Chemical	Manufacturing	paul.ryan@dowchemical.com	425-555-3374	Content marketing	Randy Owens	Writer	228-555-9507	20	$75	$1,500
24	Liberty Mutual	Banking	martha.norris@libertymutual.com	733-555-7659	Social media	Linda Herrera	Social Media Manag	320-555-4792	20	$80	$1,600
25	Kroger	Retail	martin.brewer@kroger.com	452-555-1998	Social media	Linda Herrera	Social Media Manag	320-555-4792	20	$60	$1,200
26	21st Century Fox	Entertainment	stephanie.mcdaniel@21stcentury	640-555-6958	Content marketing	Diana Pierce	Senior Writer	318-555-0419	30	$75	$2,250
27	Liberty Mutual	Banking	martha.norris@libertymutual.com	733-555-7659	Content marketing	Catherine Green	Writer	714-555-9364	15	$50	$750

Database

- collection of organized data, information and records

- creating/making a list of classmates, list of relatives, list of friends, telephone directory and so on; you're actually generating a database

- **Structural model** defines how data is organized in a database and also determine the set of operations that can be performed on a data (e.g. relational model)
- **Computer database** are those data/information stored in the computer
- **Database software** allows the user to collect, edit, delete and organize information for easy manage, update and access of the user
- **Microsoft Access** is an example of database software

Types of databases

OPERATIONAL DATABASE

- Dynamic database that is used by any organization in its day to day operation
- Use to collect, maintain, modify and delete data
- e.g. *inventory database*

ANALYTICAL DATABASE

- Static database, wherein data is rarely modified
- Use to store and track historical data to make long term projections and analysis
- e.g. *data on global temperature to determine the effects off global warming in a certain areas for a period of time*

Is the color of blood, and because of this it has historically been associated with sacrifice, danger and courage.

Usecases

ENTITY

- is the term given to the category of information in database
  - *A car dealer would be interested in entities such as **clients, cars and salesmen.***
  - *A school would be interested in **students, faculty and classes***
  - *A bookstore would be interested in **customers, stores, branches and contractors***

Tables

A database ***table*** is a collection of rows and columns that is used to organize information about a single topic. Each row within a table corresponds to a single record and contains several attributes that describe the row.

These tables are stored in databases

EmployeeID	LastName	FirstName	Department
100	Smith	Bob	IT
101	Jones	Susan	Marketing
102	Adams	John	Finance

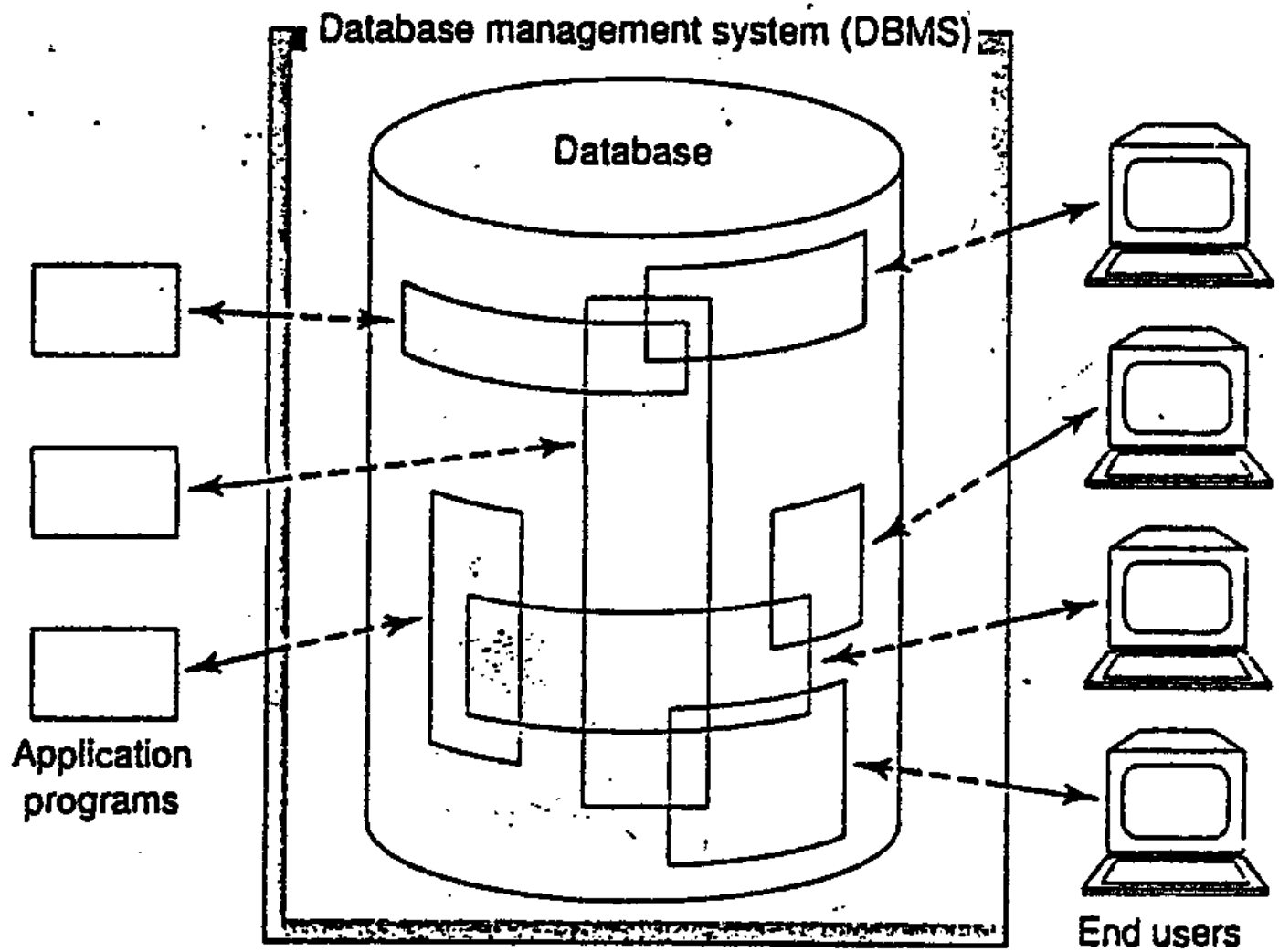


Fig. 1.4 Simplified picture of a database system

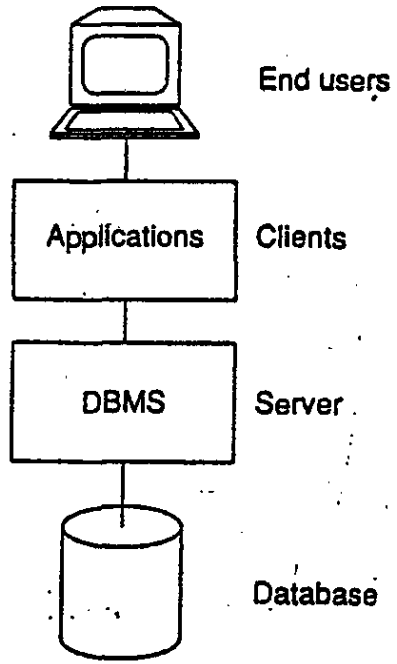


Fig. 2.5 Client/server architecture

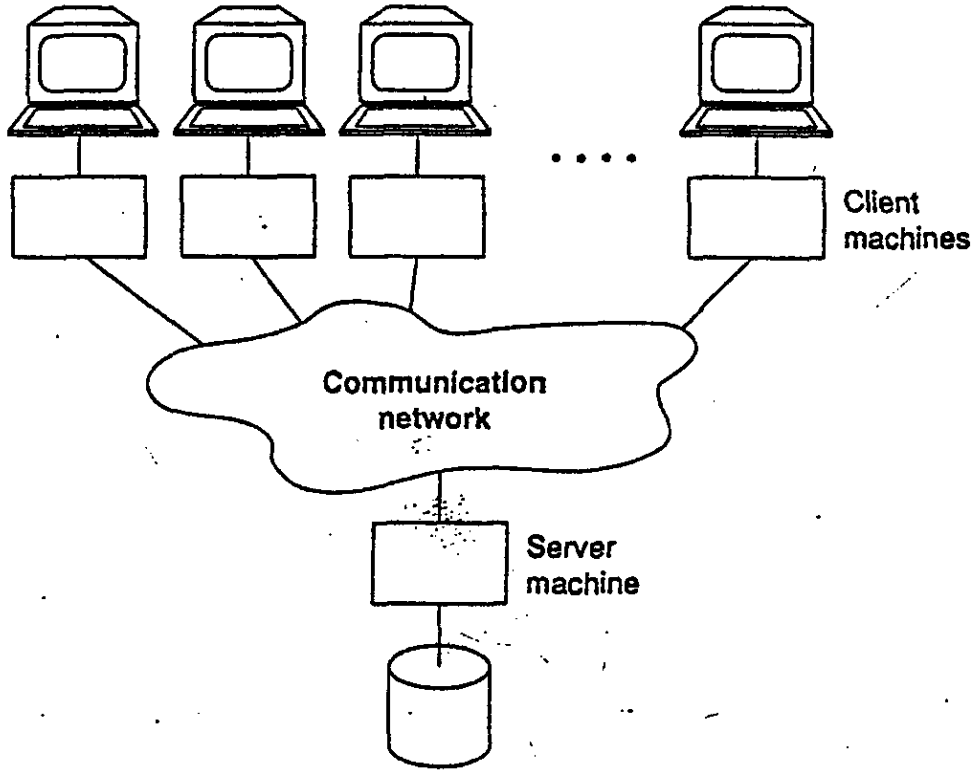


Fig. 2.7 One server machine, many client machines

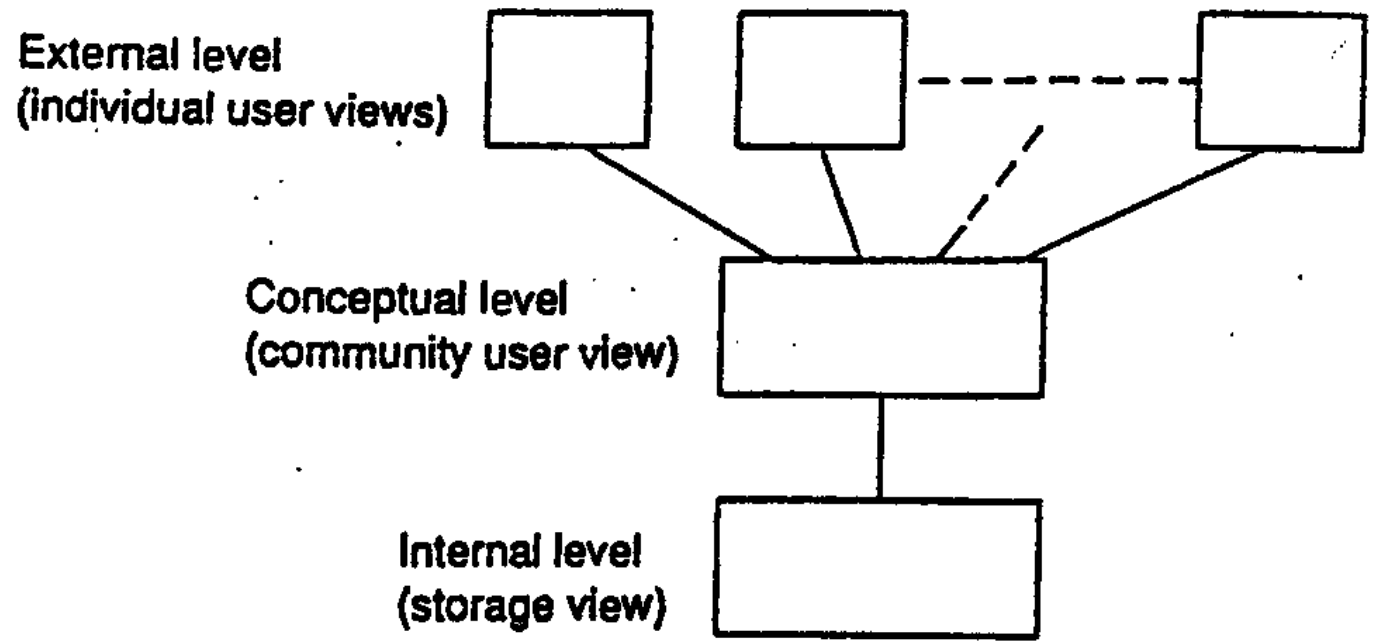


Fig. 2.1 The three levels of the architecture

Data Model

- A data model is a conceptual representation of the data structures that are required by a database.
- To use a common analogy, the data model is equivalent to an architect's building plans.
- A data model is independent of hardware or software constraints.

Type of data models

- Flat data model
- Entity relationship model
- Relation model
- Record base model
- Network model
- Hierarchical model
- Object oriented data model
- Object relation model

Flat data model

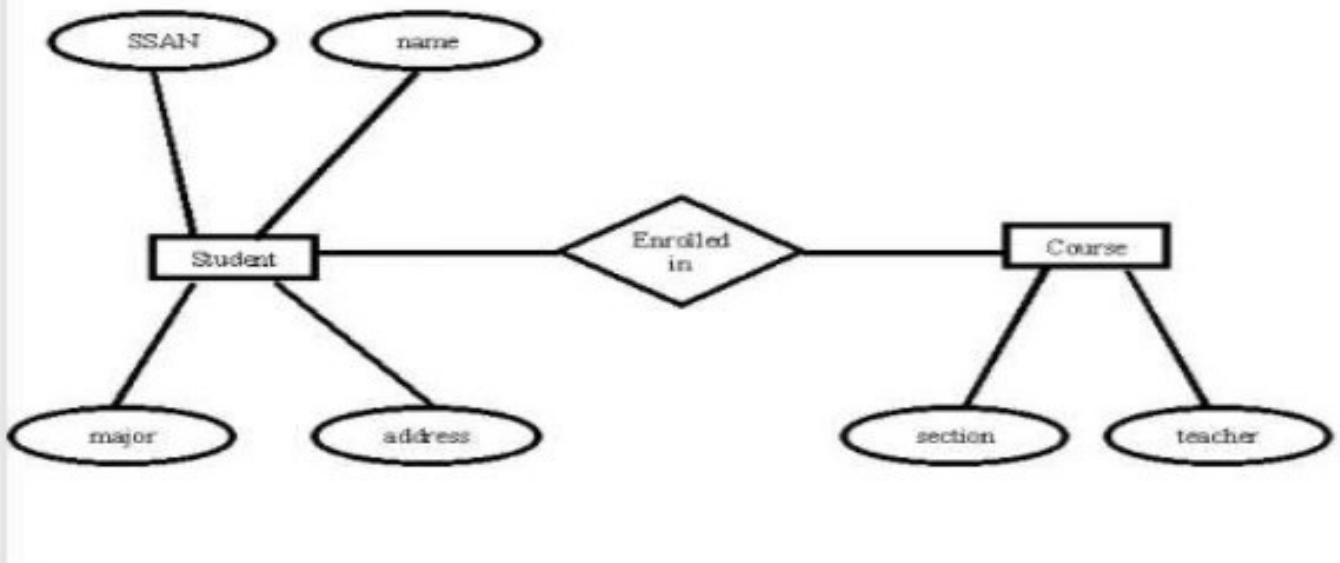
- Flat data model is the first and foremost introduced model and in this all the data used is kept in the same plane.

Roll No	Name	Course
5482	Mark	Web Designing
5486	Steve	Java
5496	Smith	Oracle

Flat Data Model

Entity Relationship Data Model

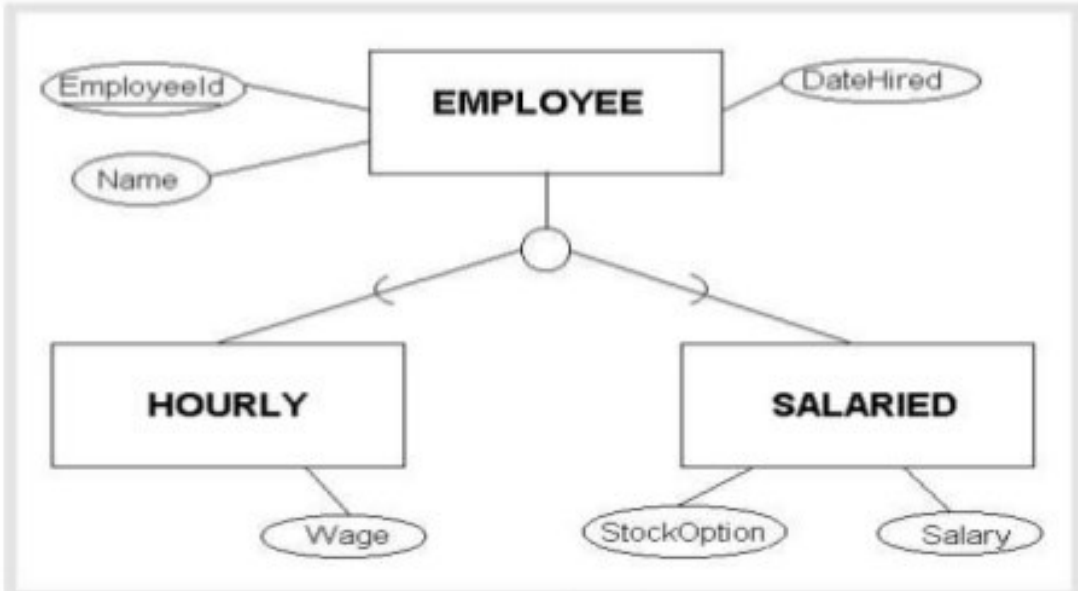
- Entity relationship model is based on the notion of the real world entities and their relationships.



Entity Relationship Model

Relational Data Model

- Relational model is the most popular model and the most extensively used model. In this model the data can be stored in the tables and this storing is called as relation. Each row in a relation contains unique value and it is called as tuple, each column contains value from same domain and it is called as attribute



Relational Model

Relational databases

A ***relational database*** a collection of tables of data all of which are formally described and organized according to the relational model. Each table must identify a column or group of columns, called the ***PRIMARY KEY***, to uniquely identify each row

Relational DBMS

- Edgar F. Codd at IBM invented the relational database in 1970. Called Father of RDBMS.
- The main elements of RDBMS are based on Codd's 13 rules for a relational system.
- Tables (or relations) are related to each other by sharing common characteristics

Codd's Rules

0	Foundation Rule
1	Information Rule
2	Guaranteed Access
3	Systematic treatment of null values
4	Active online Catalogue
5	Powerful & well structured language
6	View Updation Rule
7	Relational level operations
8	Physical Data independence
9	Logical Data independence
10	Integrity Independence
11	Distribution independence
12	Non-subversion rule

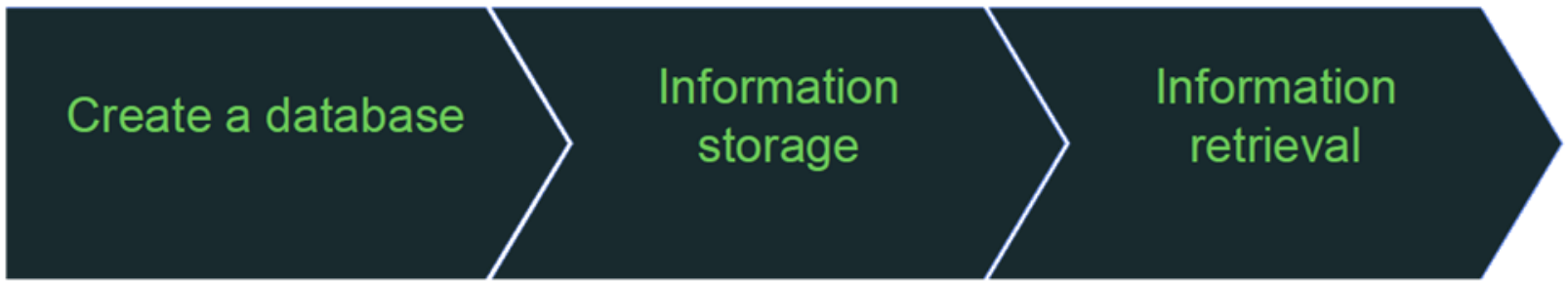
RDBMS

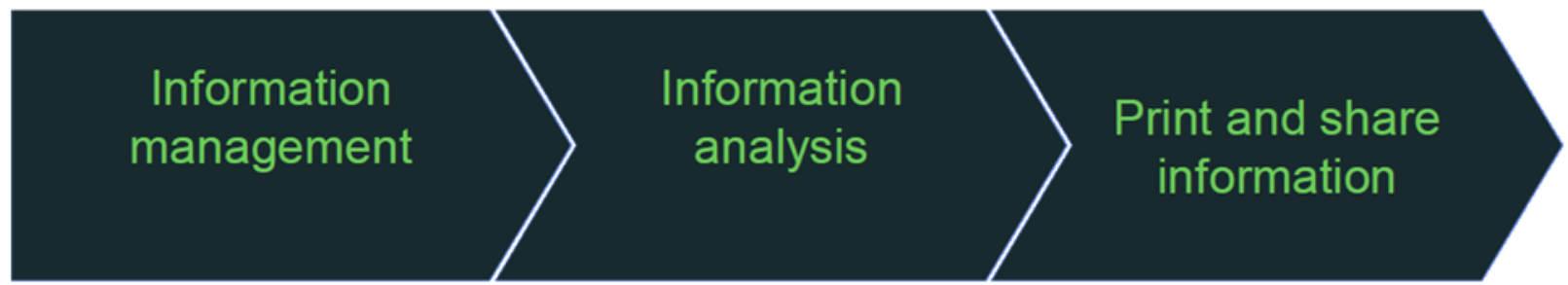
A database management system that stores data in the form of **related tables** is called Relational Database Management System.

The goal of RDBMS is to make data easy to store & retrieve

Relational databases help solve problems as they are designed to create tables & then combine the information in interesting ways to create valid information.

things we can do with RDBMS





RDBMS

Typical RDBMS include

Microsoft Access

Microsoft SQL Server

Sybase (The forerunner of Microsoft SQL Server)

IBM DB2

Oracle

Ingres

MySQL

Postgresql etc

RDBMS

Relation Instance:- snapshot of DB

Example:

branch-name	*branch-city*	*assets*
Brighton	Brooklyn	7100000
Downtown	Brooklyn	9000000
Mianus	Horseneck	400000
North Town	Rye	3700000
Perryridge	Horseneck	1700000
Pownal	Bennington	300000
Redwood	Palo Alto	2100000
Round Hill	Horseneck	8000000

Schema :- Logical design of DB.

Example:

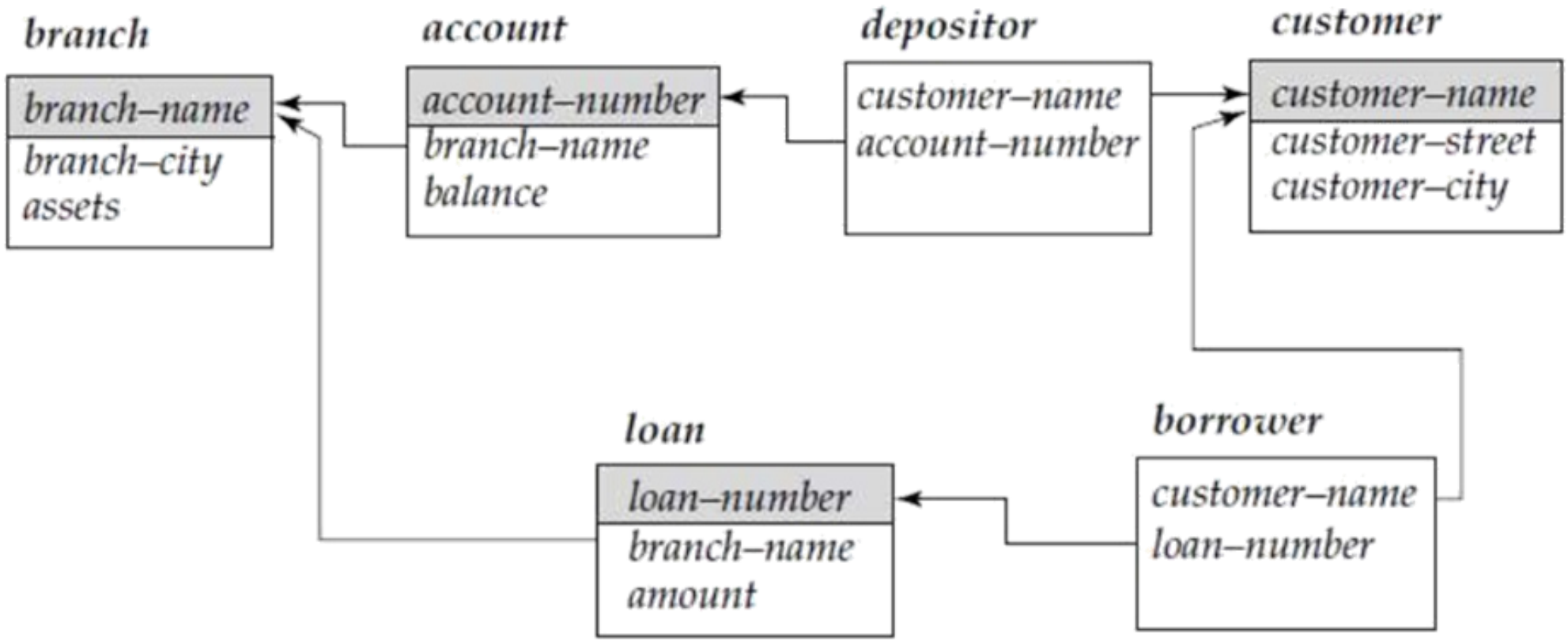
Account-schema = (account-number, branch-name, balance)

Branch-schema = (branch-name, branch-city, assets)

Customer-schema = (customer-name, customer-street, customer-city)

RDBMS

Schema Diagram



Schema diagram for the banking enterprise.

Database operations

by Angel Marchev, Jr.

Normalizing a database

Normalization the process of organizing data in a database that includes creating tables and establishing relationships between the tables

Process is used to help eliminate redundant data

Five ***normalization forms*** (NFs)

- 1NF: Eliminate Repeating Groups
- 2NF: Eliminate Redundant Data
- 3NF: Eliminate Columns Not Dependent on Key
- 4NF: Isolate Independent Multiple Relationships
- 5NF: Isolate Semantically Related Multiple Relationships

Example DB normalization

- db-operations-sample.xlsx

First normal form (1NF)

The ***first normal form*** means the data is in an entity format, which means the following conditions have been met:

- Eliminate repeating groups in individual tables
- Create separate table for each set of related data
- Identify each set of related data with primary key

Do not use multiple fields in a single table to store similar data

Example of normalization

Un-normalized table

Student#	Advisor	Adv-Room	Class1	Class2	Class3
1022	Jones	412	101-07	143-01	159-02
4123	Smith	216	201-01	211-02	214-01

First Normal Form: No Repeating Groups

Student#	Advisor	Adv-Room	Class#
1022	Jones	412	101-07
1022	Jones	412	143-01
1022	Jones	412	159-02
4123	Smith	216	201-01
4123	Smith	216	211-02
4123	Smith	216	214-01

Second normal form (2NF)

The ***second normal form*** ensures each attribute describes the entity

Create separate tables for sets of values that apply to multiple records

Relate these tables with a foreign key

Records should not depend on anything other than a table's primary key, including a compound key if necessary.

Example of normalization

Second Normal Form: eliminate redundant data

Students:

Student#	Advisor	Adv-Room
1022	Jones	412
4123	Smith	216

Registration:

Student#	Class#
1022	101-07
1022	143-01
1022	159-02
4123	201-01
4123	211-02
4123	214-01

Third normal form (3NF)

The ***third normal form*** checks for transitive dependencies.

Eliminate fields that do not depend on the key

Values that are not part of the record's key do not belong in the table

In general if the contents of a group of fields apply to more than a single record, put those fields in a separate table

Example of normalization

Third Normal Form: eliminate data not dependent on the key

Students:

Student#	Advisor
1022	Jones
4123	Smith

Faculty:

Name	Room	Dept
Jones	412	42
Smith	216	42

Registration:

Student#	Class#
1022	101-07
1022	143-01
1022	159-02
4123	201-01
4123	211-02
4123	214-01

Other normalization forms

The fourth normal form is also called the Boyce Codd Normal Form (BCNF) and fifth normal form exists, but are rarely considered in practical design

Disregarding these two additional normalization rules may result in a less than perfect database design but shouldn't affect functionality

Referential integrity

Referential Integrity (RI) is a database concept used to ensure that the relationships between your database tables remains synchronized during data modifications.

RI can be used to ensure the data is clean, may be helpful in optimizing your database environment and can assist in early detection of errors.

A combination of **PRIMARY KEY** and **FOREIGN KEY** constraints can be used to help enforce referential integrity of your database. In addition to a foreign key referencing a primary key constraint, a foreign key can also reference a **UNIQUE** constraint to help maintain referential integrity.

Triggers can also be used to enforce referential integrity, however being triggers require code they don't execute as quickly as table properties such as a primary key constraint.

Methods for enforcing referential integrity

There are several methods available in SQL Server to help maintain database integrity:

- Primary key constraint
- Foreign key constraint
- Unique constraint
- Indexes
- Triggers

Any of these methods can be created as a ***composite key*** which is an index or constraint created using more than one column. It may be necessary to use more than one column to create a unique value for each row in a table.

PRIMARY KEY constraint

An important concept of designing a database table is the use of a ***PRIMARY KEY*** — an attribute or set of attributes used to uniquely identify each row

A table can only have one primary key which is created using a primary key constraint and enforced by creating a unique index on the primary key columns

A column that participates in the primary key constraint cannot accept null values

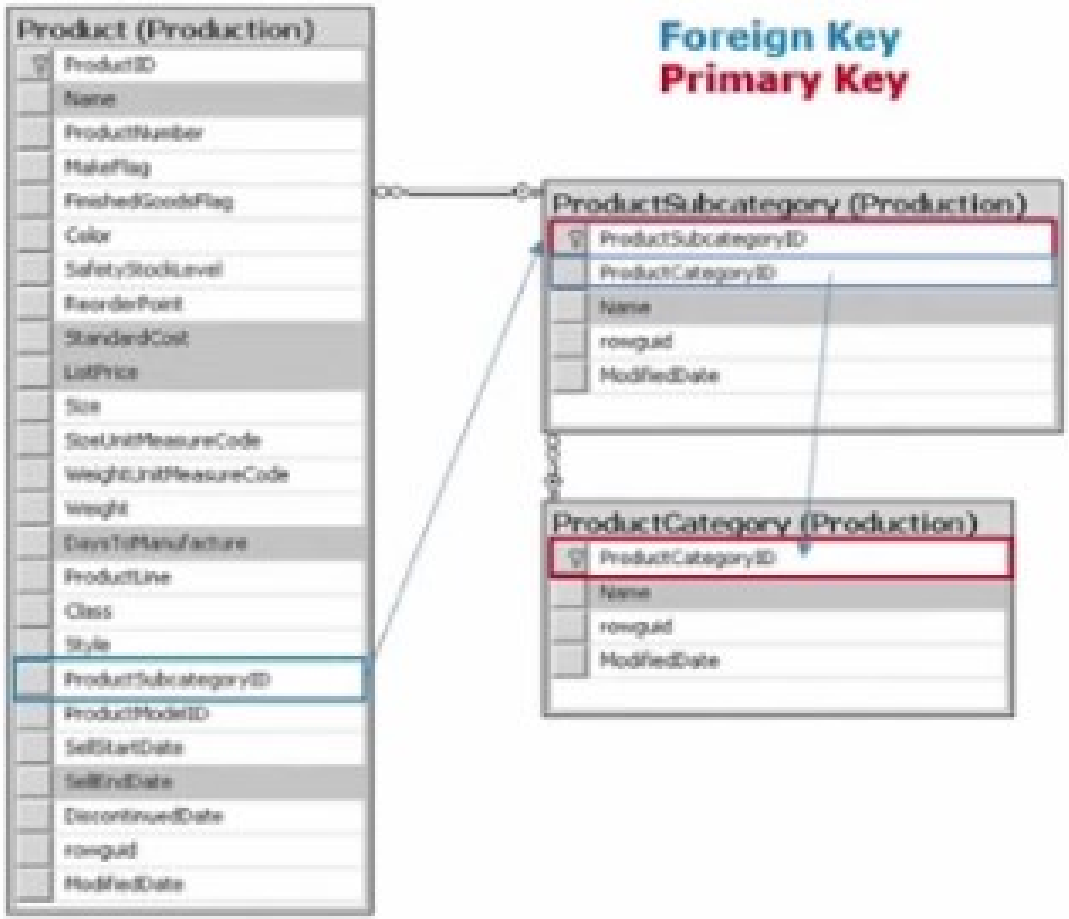
FOREIGN KEY constraint

A ***FOREIGN KEY*** is a column or combination of columns that are used to establish a link between data in two tables. The columns used to create the primary key in one table are also used to create the foreign key constraint and can be used to reference data in the same table or in another table

A foreign key does not have to reference a primary key, it can be defined to reference a unique constraint in either the same table or in another table

A column that participates in the foreign key constraint can accept null values, but if it contains a null value, the verification process is skipped.

Relational structure with keys



Primary Key Constraint

- A primary key constraint refers to a **single table**
- It identifies a subset of columns as **key columns**
- Fixing values for key columns must **identify** row
- **No two rows** have same values in key columns

Defining Example Schema

Foreign Key Constraint

- A foreign key constraint links **two tables**
- Identifies set of **foreign key columns** in table 1
- Maps foreign key columns to **primary key** of table 2
- Values in foreign key column **must appear** as primary key
- Maps **each row** in table 1 to a row from table 2

Defining Example Schema

Summary

Primary key constraint— an attribute or set of attributes used to uniquely identify each row

Foreign key constraint – a column or combination of columns used to establish a link between data in two tables

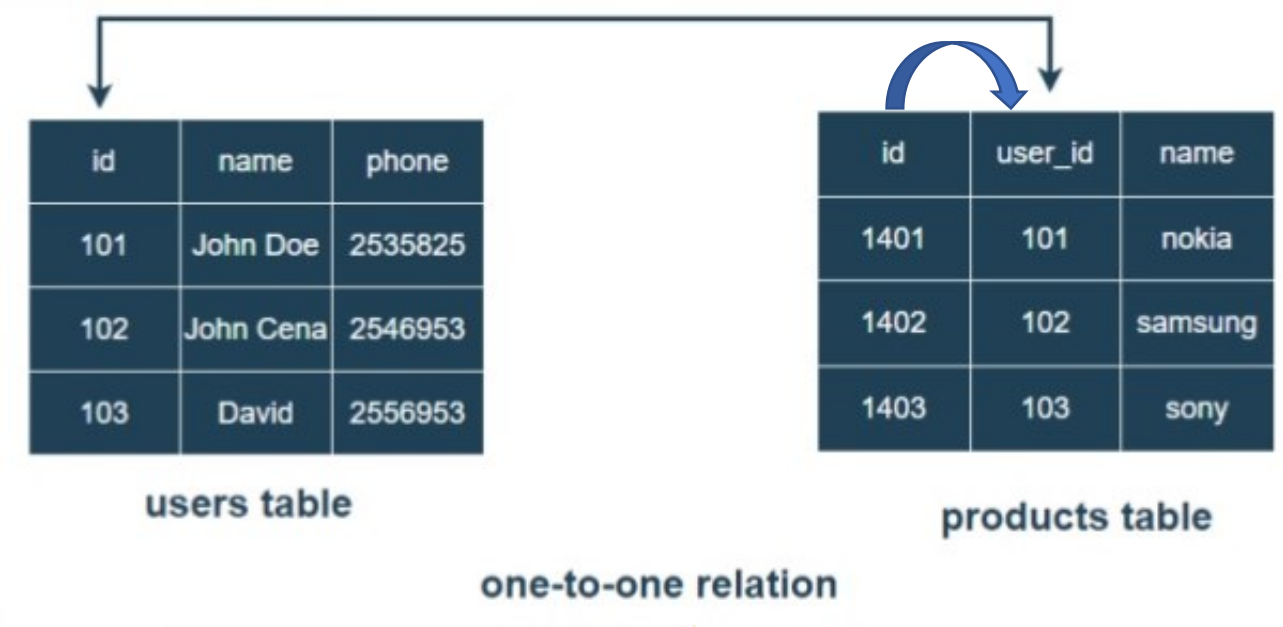
Unique constraint - allows you to enforce uniqueness in columns other than the primary key

Unique Index - ensures the index key contains no duplicate values and that every row in the table or view is unique in someway

Types of Relationship

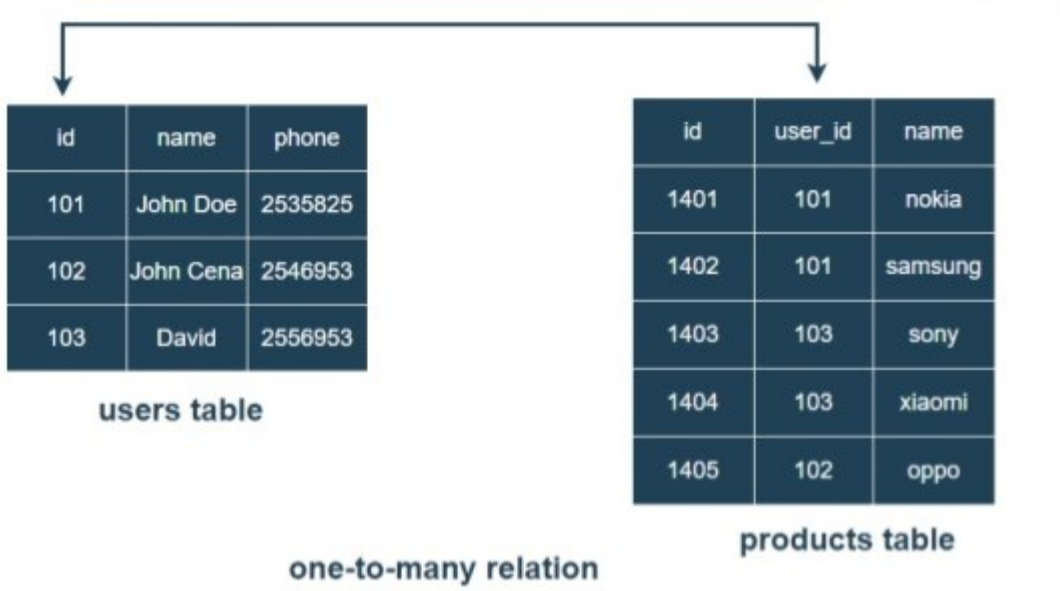
1. One-to-One relationship

- Exists between tables when only one record of the first table is related to only one record to a second table, and only one record of the second table is related to only one record to the first table



2. One-to-Many relationships

- Exists between tables when one record of the first table can be related to one or more records to a second table, but only one record from the second table can be related to a single record in the first table
- Most common relationship used to avoid or eliminate duplicate and redundant data



id	name	phone
101	John Doe	2535825
102	John Cena	2546953
103	David	2556953

users table

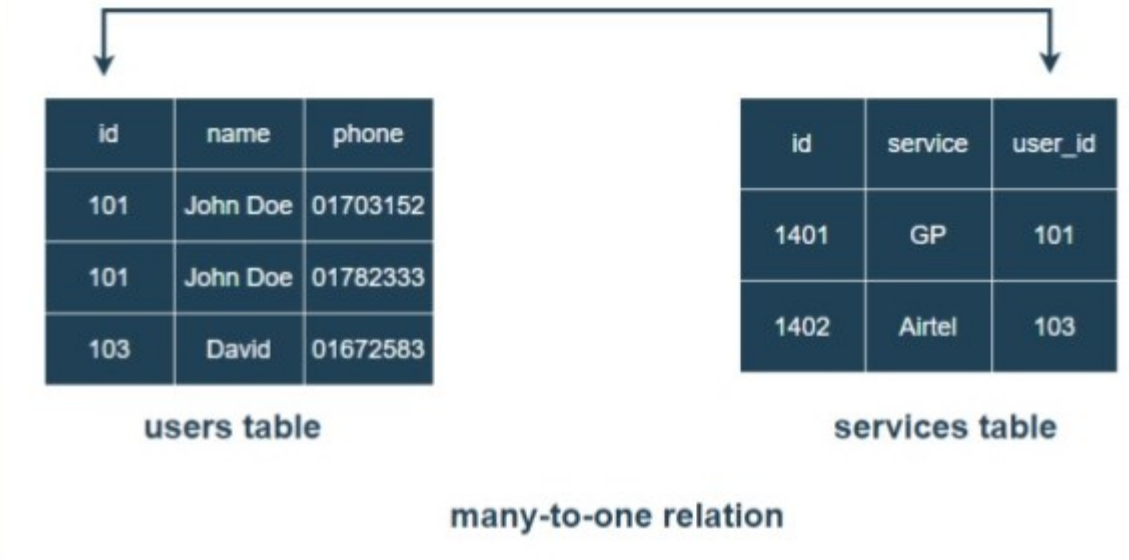
id	user_id	name
1401	101	nokia
1402	101	samsung
1403	103	sony
1404	103	xiaomi
1405	102	oppo

products table

one-to-many relation

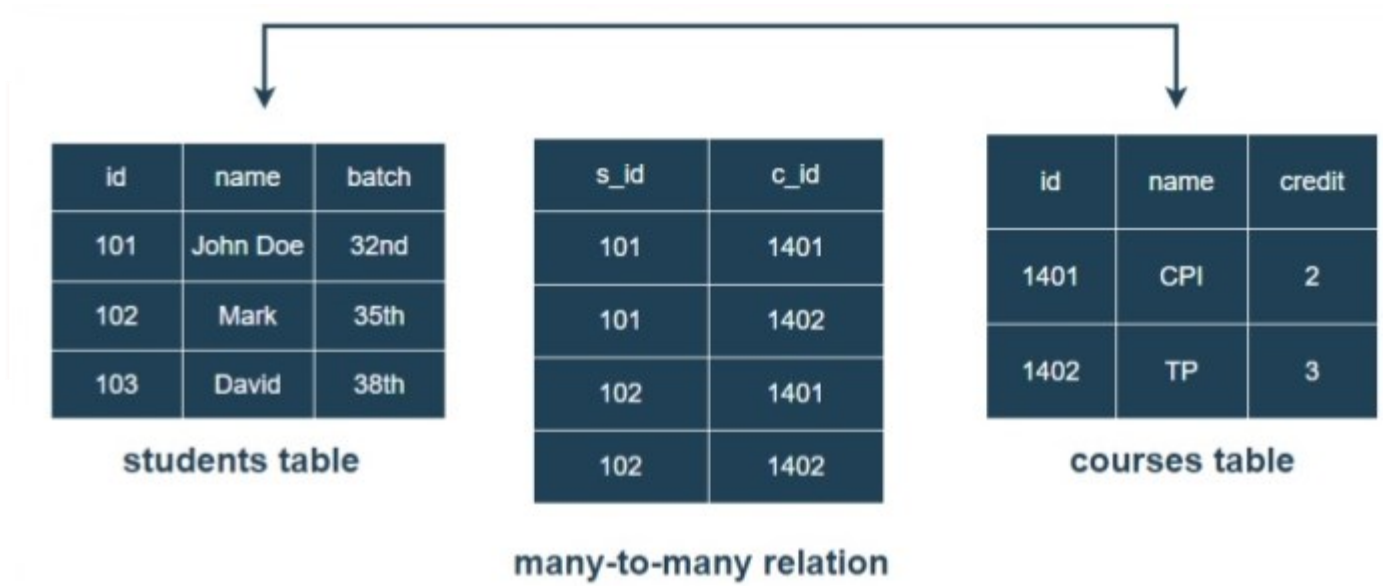
2.Many-to-one relationships

□Exists between tables when more than one record of the first table can be related to one record to a second table, but only one record from the second table can be related to a single record in the first table



3. Many-to-Many relationship

□Exists between tables when one record of the first table can be related to one or more records to a second table, and one or more records of the second table is related to one or more record s to the first table



id	name	batch
101	John Doe	32nd
102	Mark	35th
103	David	38th

students table

s_id	c_id
101	1401
101	1402
102	1401
102	1402

many-to-many relation

id	name	credit
1401	CPI	2
1402	TP	3

courses table

Data normalization

- 1) Do the task
- 2) Save the file
- 3) Submit it here (incl. your answers) :
- https://forms.gle/7YRmC4CehbdGBBby7

Exercise 3: Data normalization (homework)

- 1) Do the task
- 2) Save the file
- 3) Submit it here (incl. your answers) :
- https://forms.gle/7YRmC4CehbdGBBby7

Database Queries

by Angel Marchev, Jr.

Database objects

- **Table** – collection of rows and columns which are used to store information about single topic (entity). Each row is a record, with data about several attributes.
- **View** – virtual table consisting of records and columns from different tables. It is stored in the database as a query object, not as a table.
- **Procedure** – compiled list of instructions and queries, saved in the database to be run several times.

Database Interfaces

- **Form** – user interface to the database for inputting and outputting data.
- **Report** – formatted document containing outputs from a database.

Query – scripted request for extracting certain data from the database – with or without certain data processing.

Queries help you find and work with your data

In a well-designed database, the data that you want to present through a form or report is usually located in multiple tables. A query can pull the information from various tables and assemble it for display in the form or report. A query can either be a request for data results from your database or for action on the data, or for both. A query can give you an answer to a simple question, perform calculations, combine data from different tables, add, change, or delete data from a database. Since queries are so versatile, there are many types of queries and you would create a type of query based on the task.

Major query types	Use
Select	To retrieve data from a table or make calculations.
Action	Add, change, or delete data. Each task has a specific type of action query.

We can use Relational Algebra to fetch data from this Table(relation)

Select Name students with age less than 17

Output

ID	Name	Age
1	Akon	17
2	Bkon	19
3	Ckon	15
4	Dkon	13

Name
Ckon
Dkon

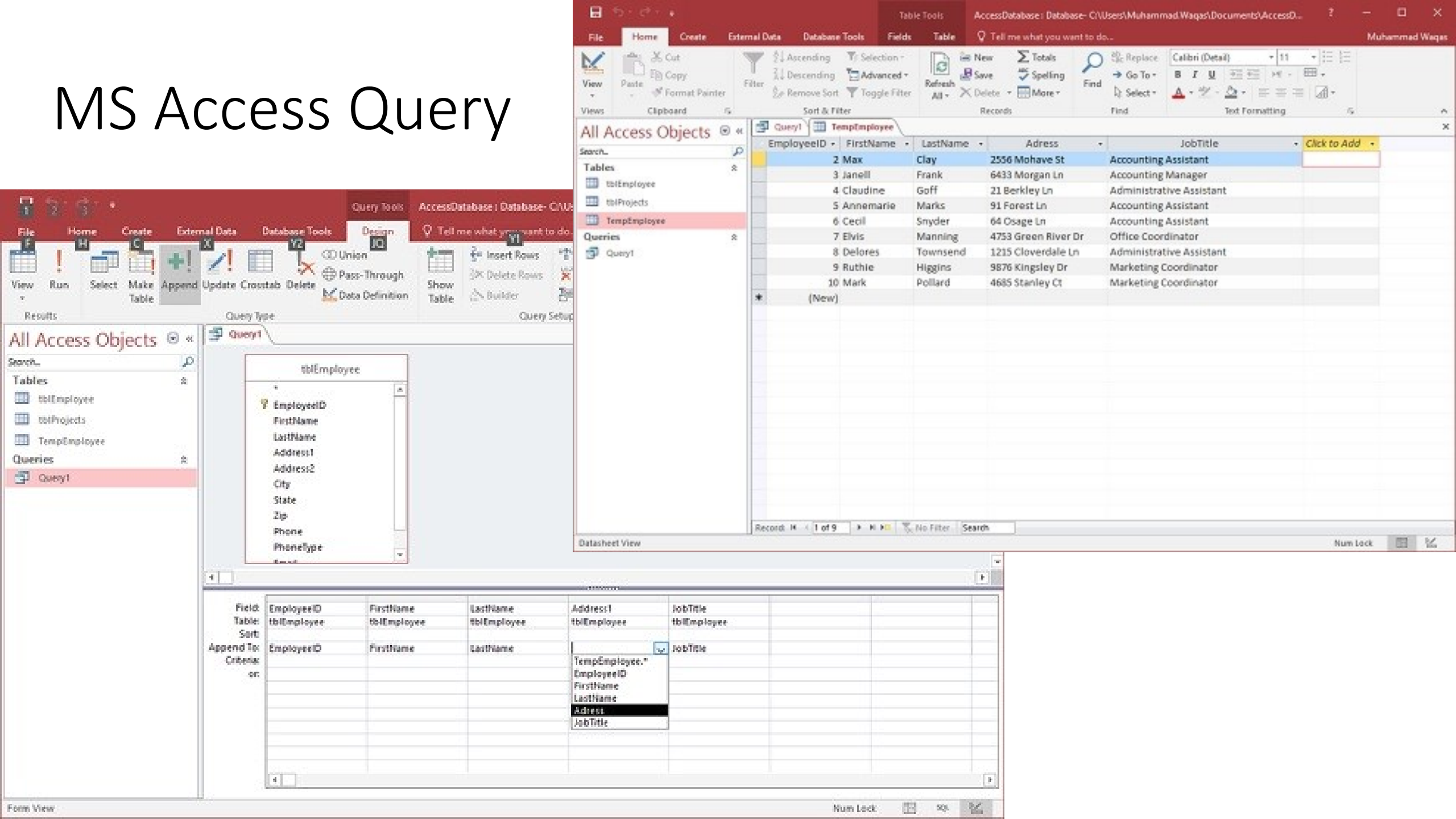
The output for query is also in form of a table(relation), with results in different columns

1. Use `GROUP BY` to combine rows
2. Use `FILTER` to pick rows per column

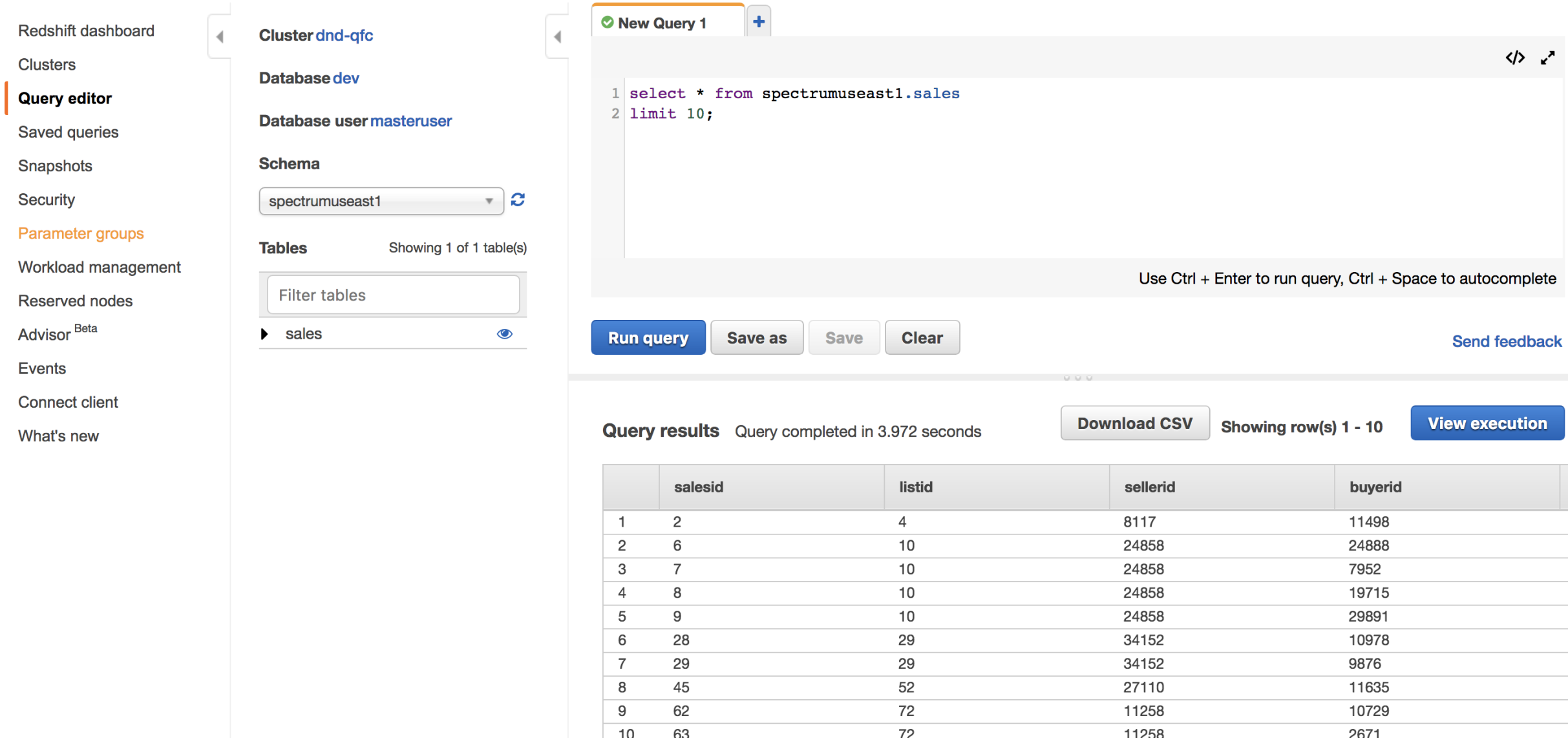
Year	Month	Revenue
2016	1	1
2016	2	23
2016	3	345
2016	...	...
2016	12	1234

Year	Jan	Feb	Mar	...	Dec
2016	1	23	345	...	1234

MS Access Query



SQL Select Query



QUERY DESIGN

- All queries need to be deigned and should always include the following:
    - Fields
    - Tables
    - Criteria for query
    - Sort order
    - Grouping

QUERY DESIGN TEMPLATE

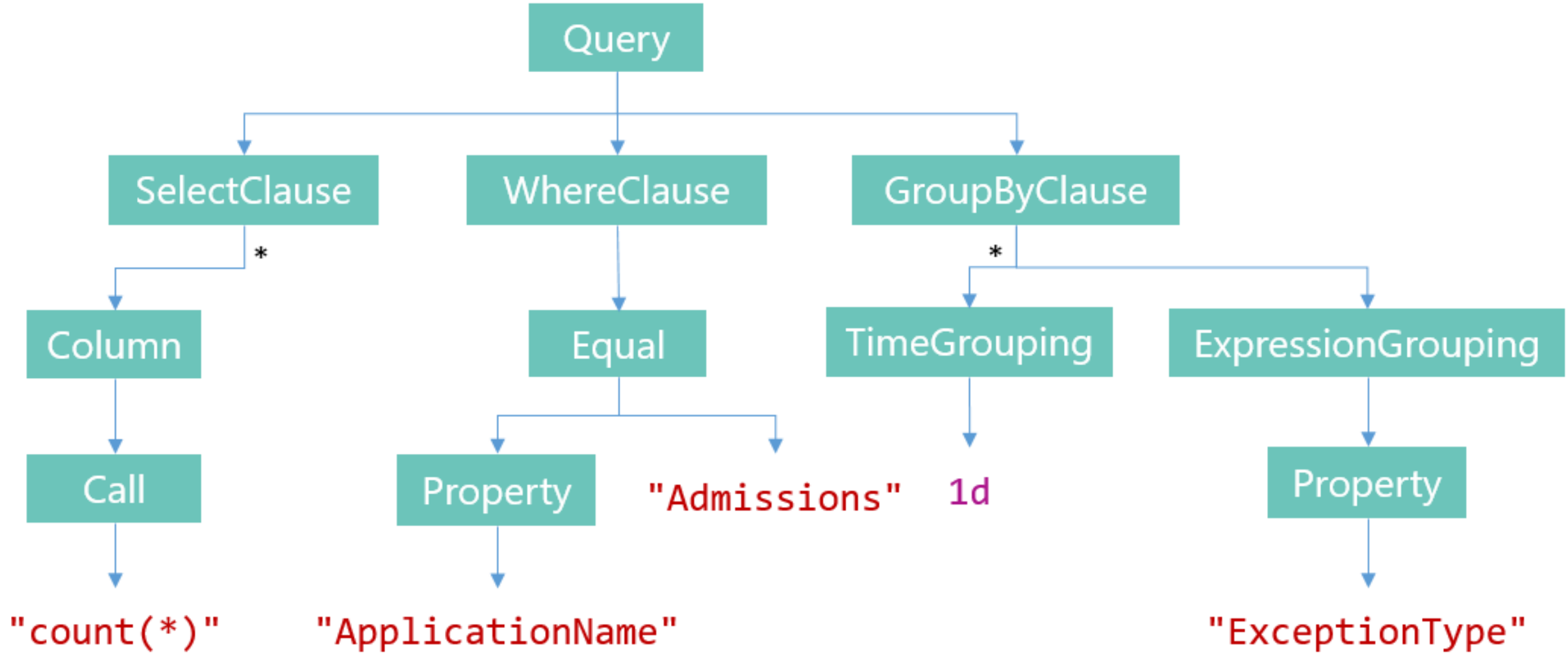
- The format for the query design table:

Field (s)	This shows the fields required for the query.
Table (s)	This shows the table each field comes from, this can be more than one.
Search Criteria	This shows the conditions required for the query.
Sort Order	If the query need to sort the answers.

EXAMPLE

- Design a query to list Computing teachers and phone numbers with a tutor class. This should be sorted in alphabetical order.

Field (s)	teacherName, teacherNumber, tutorGroup, Subject
Table (s)	Teacher
Search Criteria	Subject = Computing AND tutorGroup = True
Sort Order	Asec


Query
SelectClause
WhereClause
GroupByClause
*
*
Column
Equal
TimeGrouping
ExpressionGrouping
Call
Property
"Admissions"
1d
Property
"count(*)"
"ApplicationName"
"ExceptionType"

QUERY DESIGN

- Query designs should include the information shown below
- Depending on the query, not all elements need to be filled

Field(s) and calculation(s)	
Table(s) and query	
Search criteria	
Grouping	
Sort order	

CAR HIRE DATABASE EXAMPLE

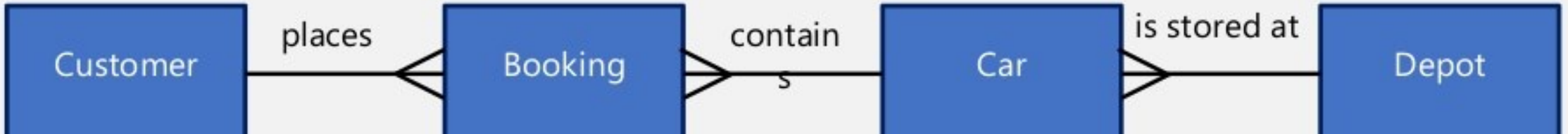
- The following SQL query designs will relate to a car hire database which consists of 4 tables:

Customer
customerID
forename
surname
telephone
houseNum
postcode

Booking
bookingRef
customerID *
registration *
daysBooked
startDate

Car
registration
make
model
colour
depotID *
automatic
dailyPrice
mileage

Depot
depotID
depotName
depotCity
onlineBooking
depotHours
employees



Database queries exercise

- https://forms.gle/B4qNS3XdQqJPacqG9

EXAMPLE 1

- Design a query to show the make, model and mileage of a car in a depot which is open all day, and where the make of car starts with 'M'.

Field(s) and calculation(s)	
Table(s) and query	
Search criteria	
Grouping	
Sort order	

EXAMPLE 1

- Design a query to show the make, model, mileage and depot of any car in a depot that is open all day and where the make of car starts with 'M'.

Field(s) and calculation(s)	make, model, mileage, depotName, depotHours
Table(s) and query	car, depot
Search criteria	depotHours = 'All Day' and model like 'M%'
Grouping	
Sort order	

EXAMPLE 2

- Design a query to show a customer's full name, booking ID, car make and model, and depot name for all customers who have 'e' as the 5th letter of their surname
- List the details in alphabetical order of the customer surname

Field(s) and calculation(s)	
Table(s) and query	
Search criteria	
Grouping	
Sort order	

EXAMPLE 2

- Design a query to show a customer's full name, booking ID, car make and model, and depot name for all customers who have 'e' as the 5th letter of their surname
- List the details in alphabetical order of the customer surname

Field(s) and calculation(s)	forename, surname, bookingRef, make, model, depotName
Table(s) and query	customer, booking, car, depot
Search criteria	surname LIKE = '_ _ _ _ e %'
Grouping	
Sort order	surname ASC