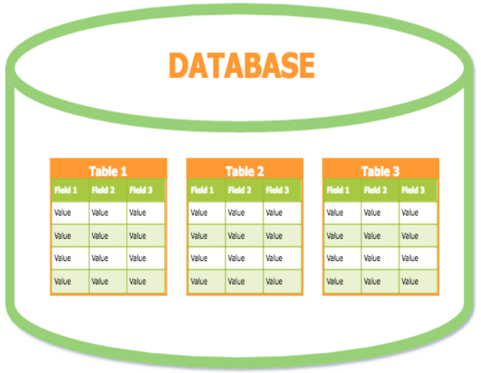
SQL Introduction (Structured Query Language)

Session 1

Agenda

- What is SQL?
- History of SQL
- Why is SQL Important?
- Popular Databases
- Why MySQL DB?
- Installation

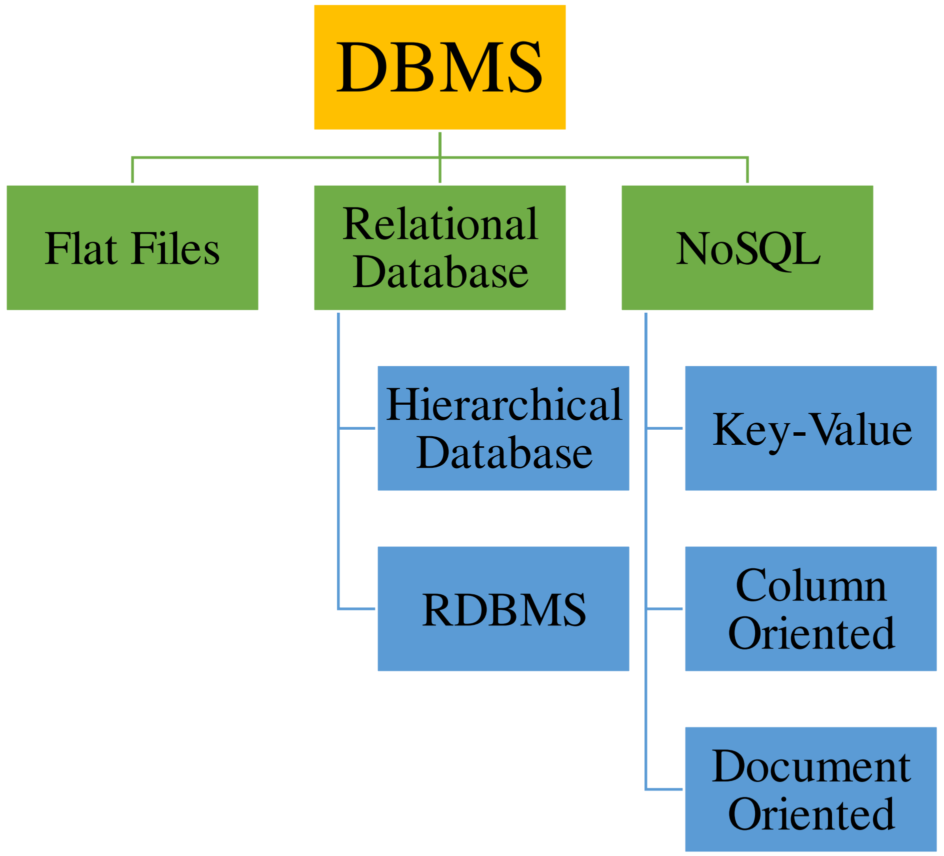
When do you need a Database?



- Databases can store very large numbers of records efficiently
- Web interfaces to data
- It is very quick and easy to find **information**
- Easy to add new **data** and to edit data
- Data changes on a regular basis
- Multiple simultaneous changes to data
- Share huge data set among many people

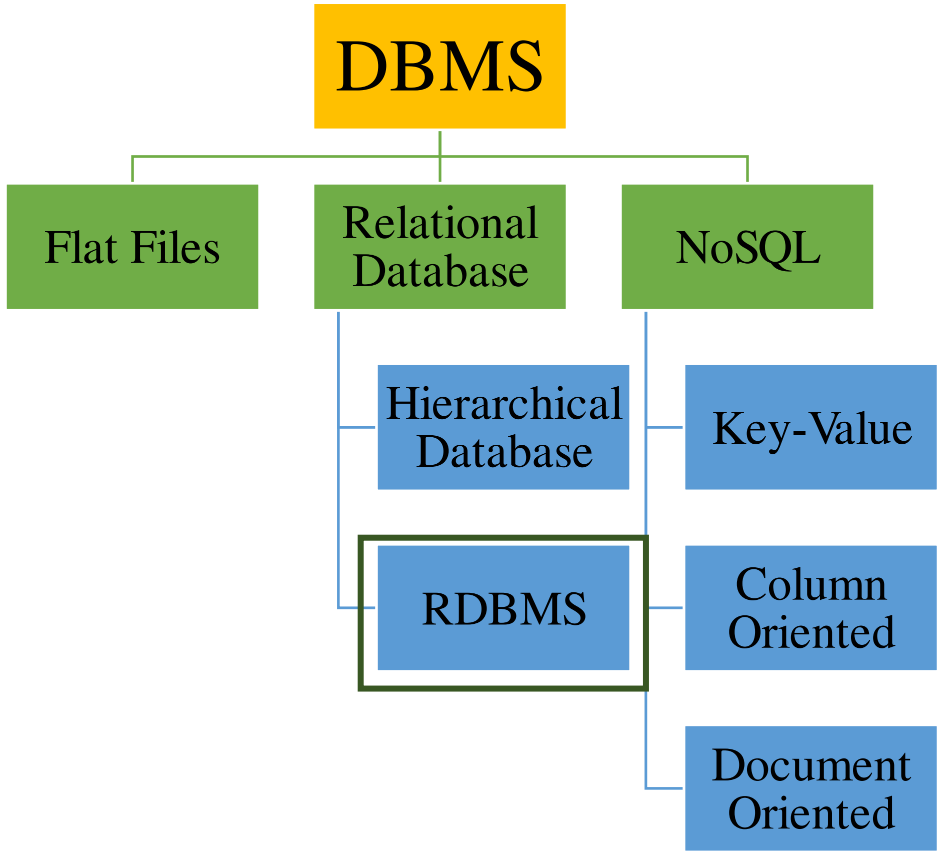
DBMS (Database Management System)

- Technically, it is software system to manage complete database



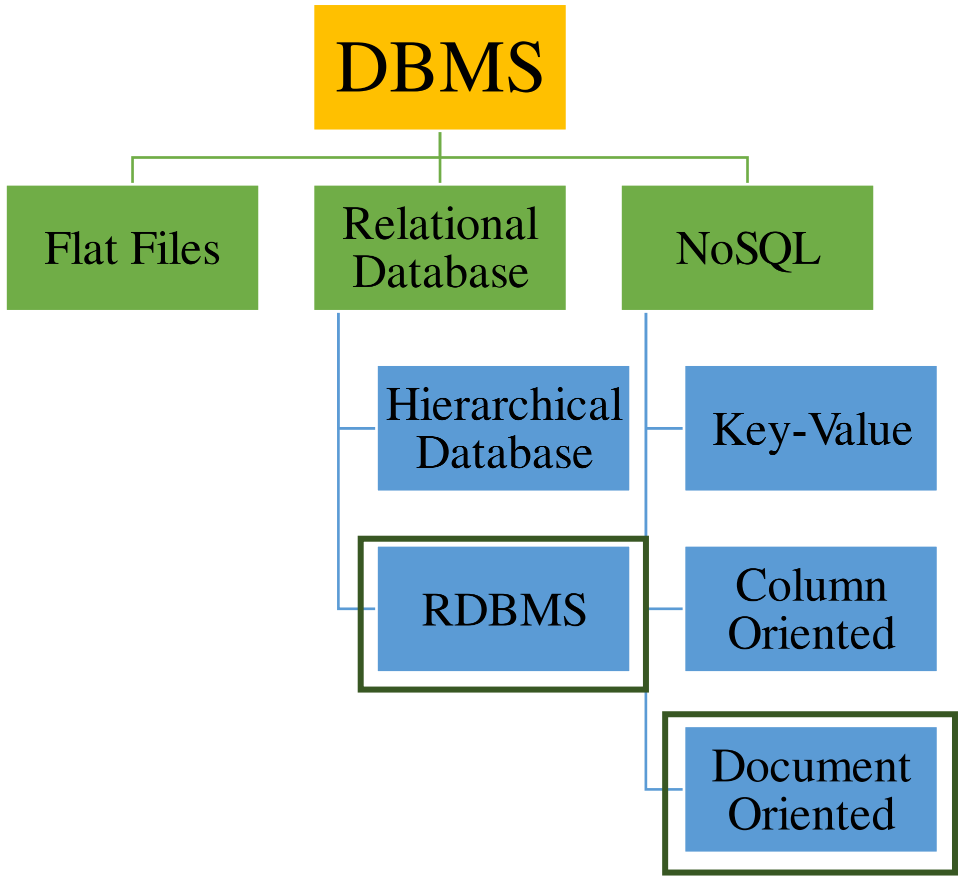
DBMS (Database Management System)

- Technically, it is software system to manage complete database



DBMS (Database Management System)

- Technically, it is software system to manage complete database



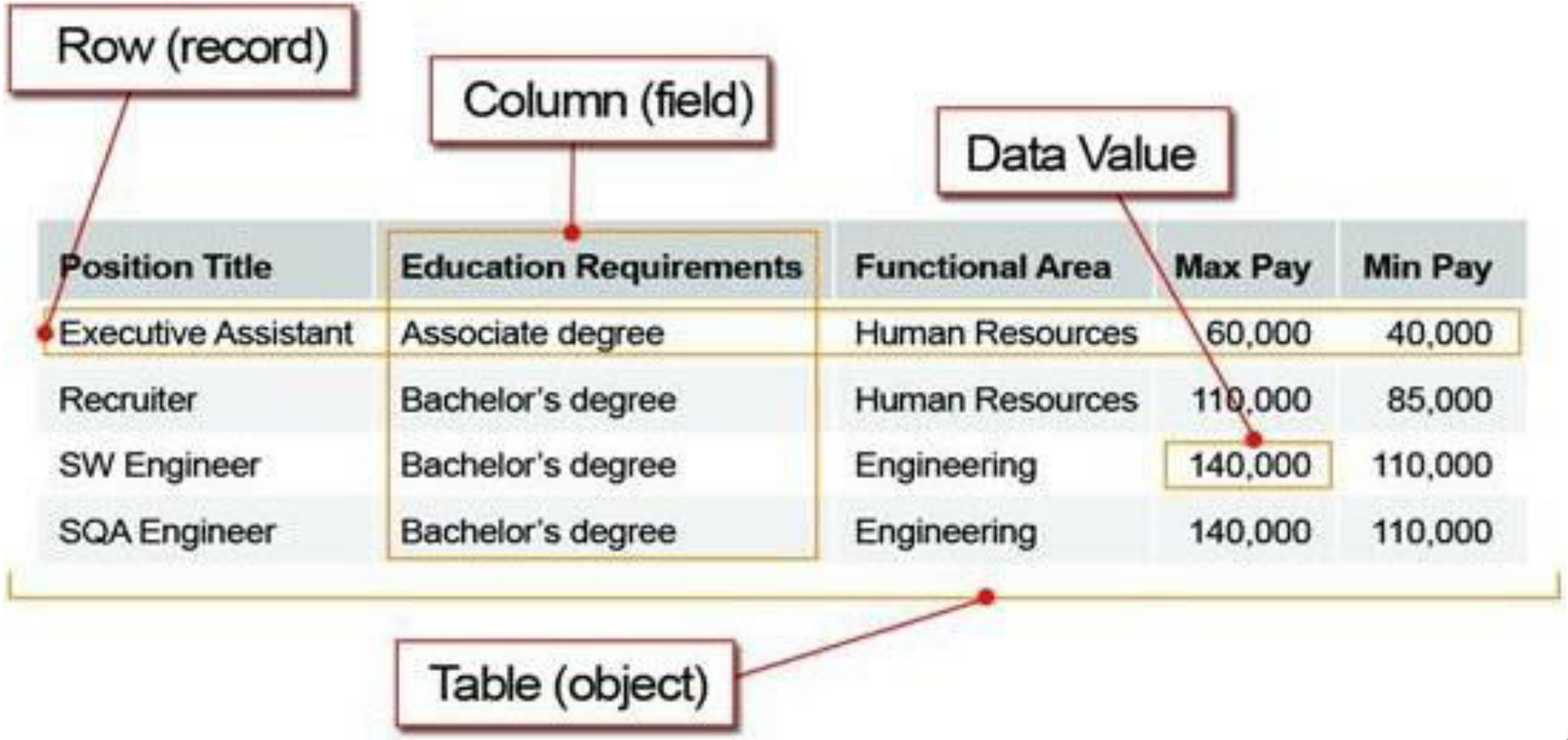
SQL

- Stands for Structured Query Language
- SQL is used to communicate with a database
- SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database
- Some common database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server

History of SQL

- The SQL programming language was first developed in the 1970s by IBM researchers Raymond Boyce and Donald Chamberlin
- Commercially released by Relational Software Inc. in 1979
- The early version of the language, which was referred to as SEQUEL (short form for Structured English Query Language)
- The name of this version was later changed from SEQUEL to SQL

Data storage example in database

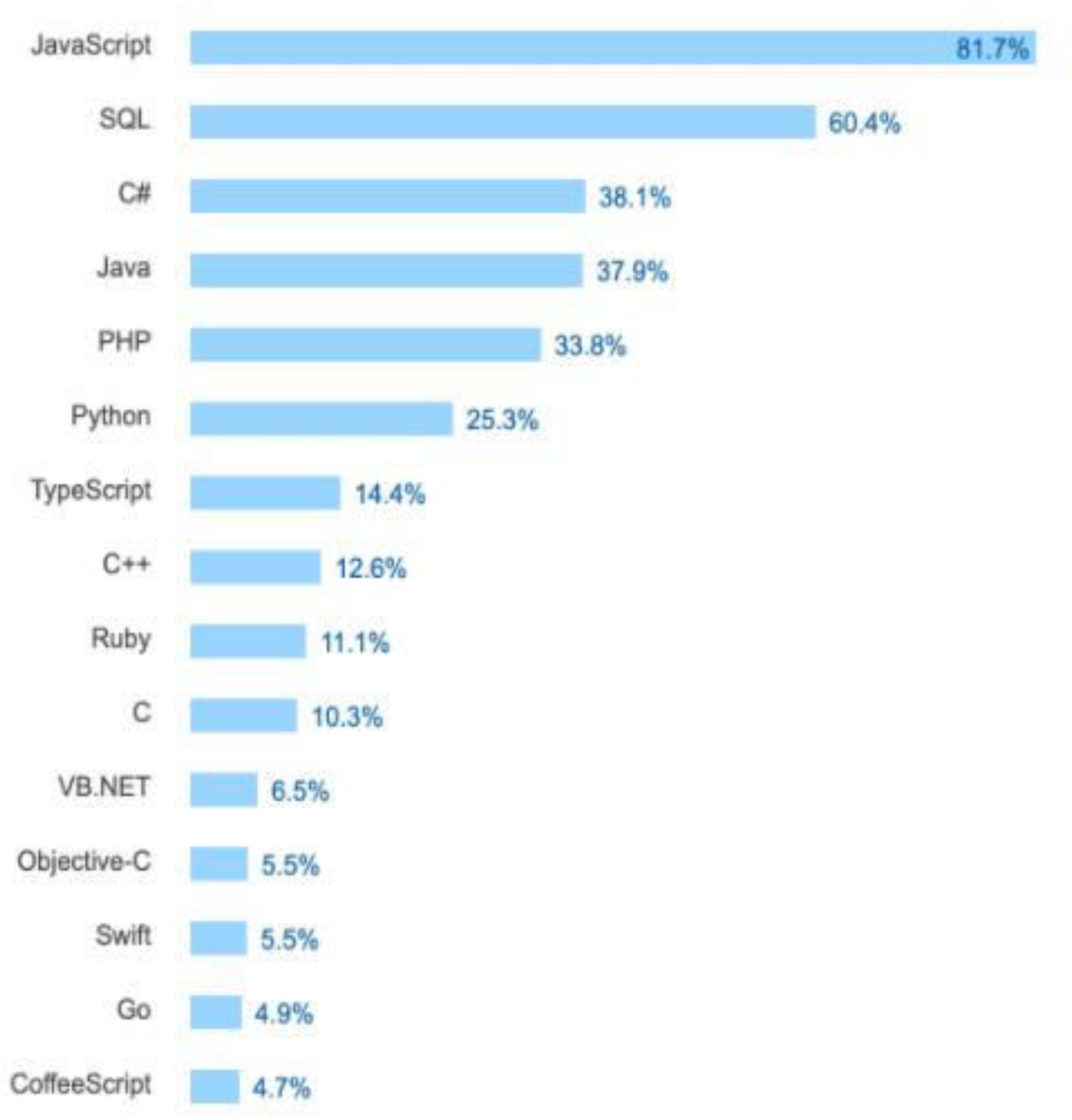


Position Title	Education Requirements	Functional Area	Max Pay	Min Pay
Executive Assistant	Associate degree	Human Resources	60,000	40,000
Recruiter	Bachelor's degree	Human Resources	110,000	85,000
SW Engineer	Bachelor's degree	Engineering	140,000	110,000
SQA Engineer	Bachelor's degree	Engineering	140,000	110,000

Why is SQL Important?

- SQL is used everywhere
- Its in high demand always
- Although there are alternatives,

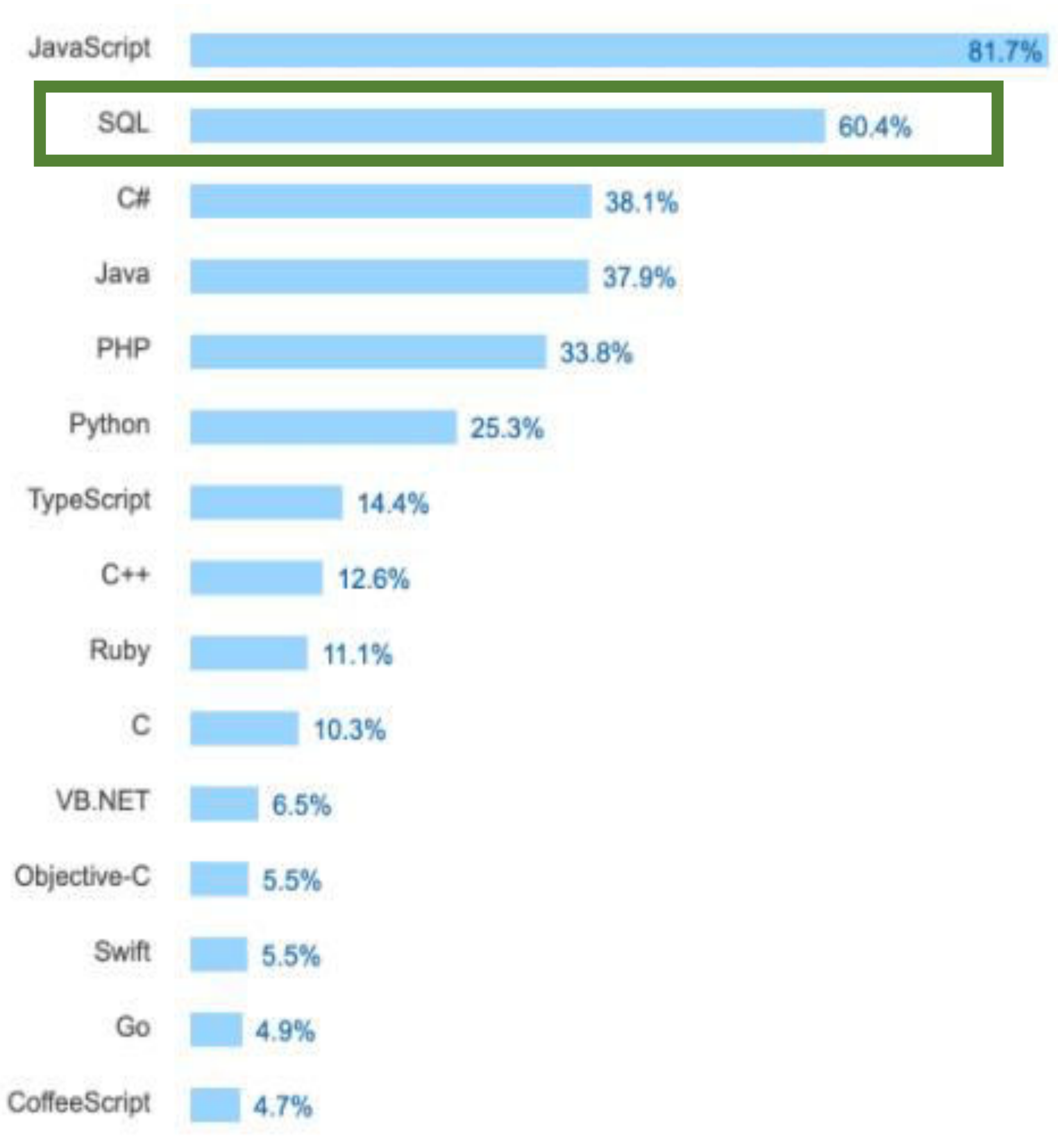
  SQL cannot be replaced
- Any analysis always starts from SQL
- Easiest way to understand data



Why is SQL Important?

- SQL is used everywhere
- Its in high demand always
- Although there are alternatives,

  SQL cannot be replaced
- Any analysis always starts from SQL
- Easiest way to understand data



OLAP vs. OLTP

- Two different types of databases and two different ways of processing data

- **OLTP - *Online transaction processing***
- **OLAP -** *Online Analytical Processing*

- OLTP (RDMS) Examples-
  - MySQL, Oracle, PostgreSQL

- OLAP (Storage) Examples -
  - Hadoop System, Hive

OTLP

- Lets us take an example –
- We have 2 person's who is actually doing a online transaction

- Person A

Person B

Account	Balance
XXXXX1234	$ 6000

Account	Balance
XXXXX5678	$ 1500

OTLP

- Lets us take an example –
- We have 2 person's who is actually doing a online transaction

- Person A

Google Pay of $ 2000

Person B

Account	Balance
XXXXX1234	$ 6000

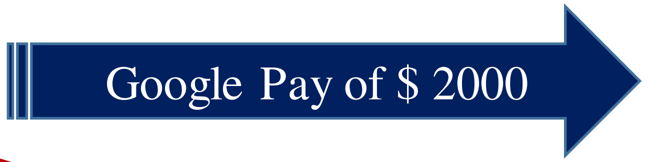
Account	Balance
XXXXX5678	$ 1500

OTLP

- Lets us take an example –
- We have 2 person's who is actually doing a online transaction

- Person A

Person B



Account	Balance
XXXXX1234	$ 6000

Account	Balance
XXXXX5678	$ 1500

OTLP

- Lets us take an example –
- We have 2 person's who is actually doing a online transaction

- Person A

Google Pay of $ 2000

Person B

Account	Balance
XXXXX1234	$ 6000

$ 4000

After Successful Transaction

Account	Balance
XXXXX5678	$ 1500

$ 3500

OLTP

- System which manages transaction-oriented applications
- Manage day-to-day an time-to-time transactions

Examples –

ATM Transaction

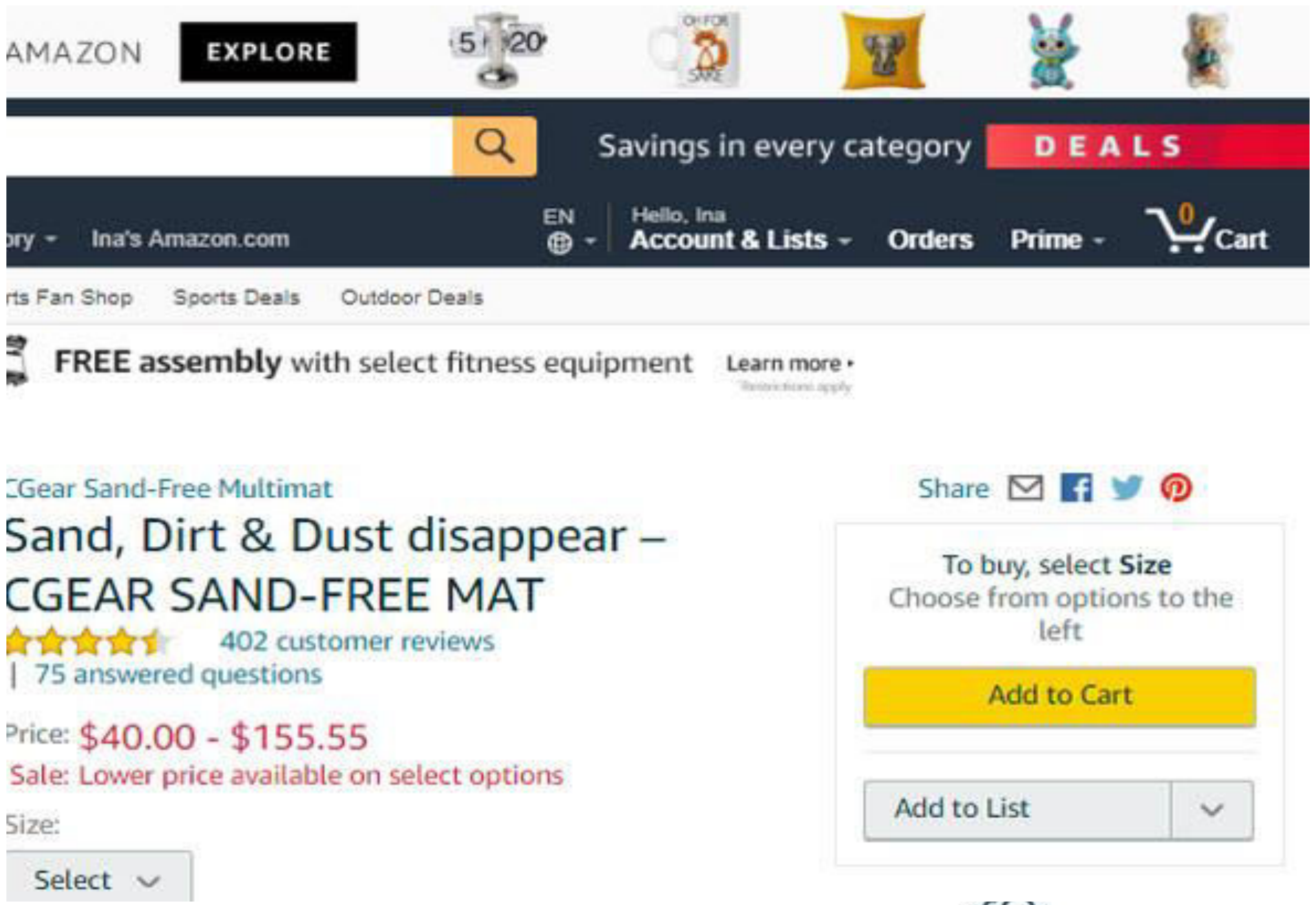
Air Ticket Booking

Online Purchase

OLAP

- Let us consider a Amazon User
- User tried to buy a product
- But something was stopping him to purchase

Now Amazon analysed his last 3 days activates and came up with some discounts



FREE assembly with select fitness equipment

CGear Sand-Free Multimat

Sand, Dirt & Dust disappear – CGEAR SAND-FREE MAT

402 customer reviews | 75 answered questions

Price: $40.00 - $155.55

Sale: Lower price available on select options

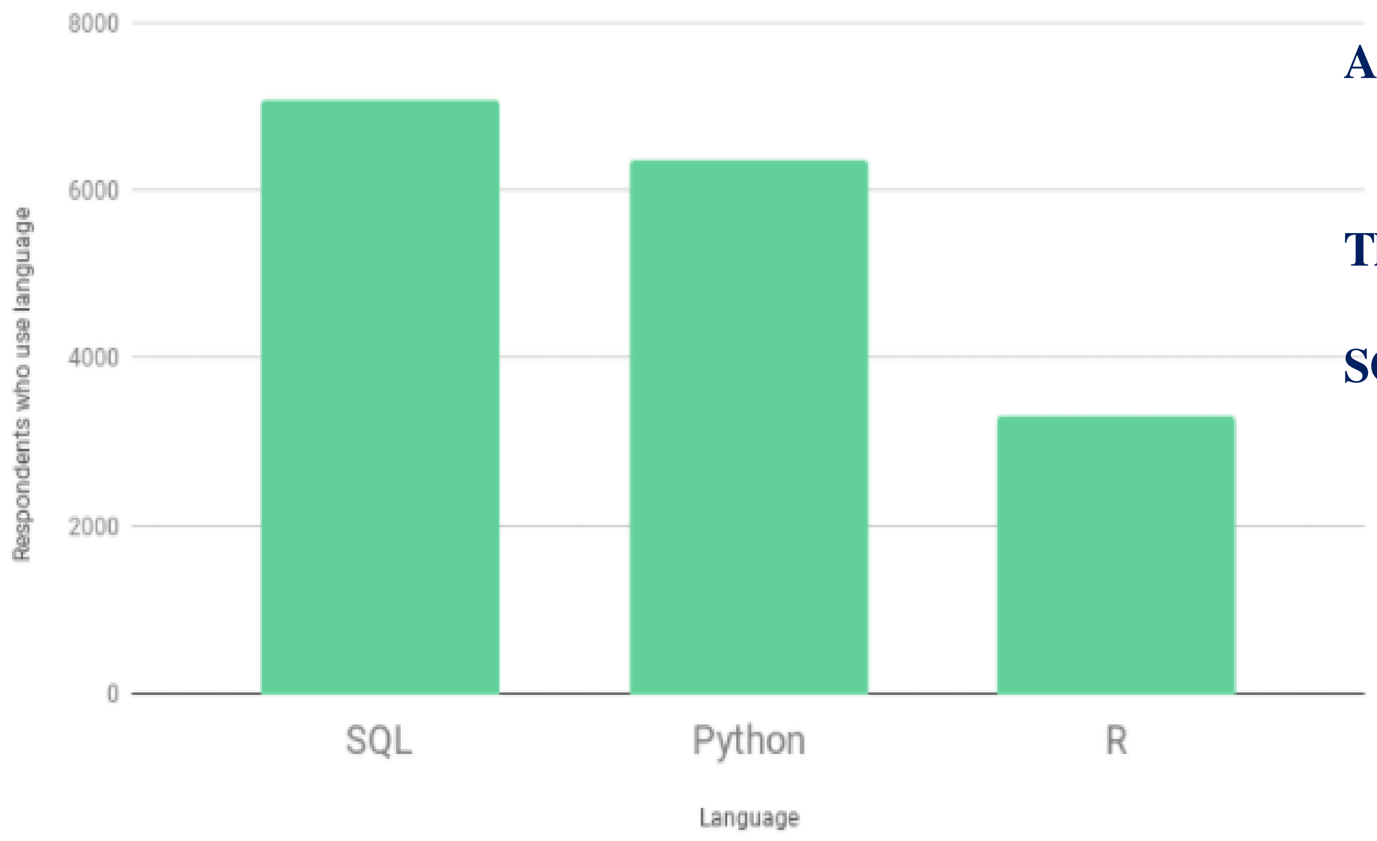
Size:

To buy, select Size
Choose from options to the left

OLTP vs. OLAP

Characteristics	OLTP	OLAP
Volume	Handles a large number of small transactions	Handles large volumes of data with complex queries
Query types	Simple standardized queries	Complex queries
Response time	Milliseconds	Minutes, or hours depending on the amount of data to process
Source	Transactions	Aggregated data from transactions
Purpose	Control and run essential business operations in real time	Plan, solve problems, support decisions, discover hidden insights

Languages used by Data Analyst / Data Scientist



About 45% of data comes from OTLP

Though the world is moving towards – Big Data

SQL will never be replaced

SQL Databases

Microsoft® SQL Server®

PostgreSQL

IBM DB2

MySQL™

ORACLE®

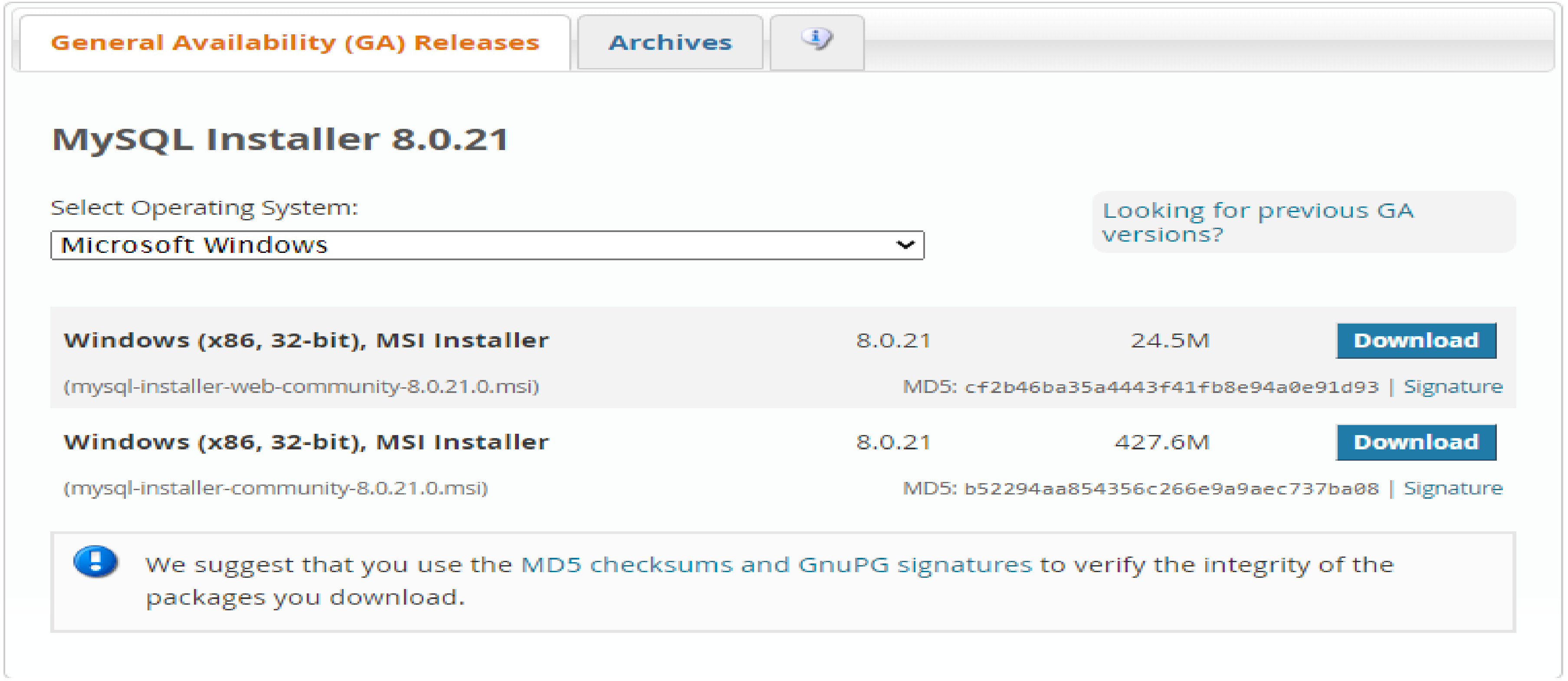
Why MySQL?

MySQL	Free (Open Source)
	Highly configurable
	High volume capabilities
	Cross platform Compatibilities
	Durability
	Reliable
	High Security
	Easy to Use

facebook
Wikipedia
The Free Encyclopedia
LinkedIn
Baidu 百度
PayPal
iStockphoto
flickr
verizon wireless
NASA
Adobe
Walmart
Save money. Live better.

MySQL Download

- Official Website - https://www.mysql.com/



MySQL Installer 8.0.21

Select Operating System:

Microsoft Windows

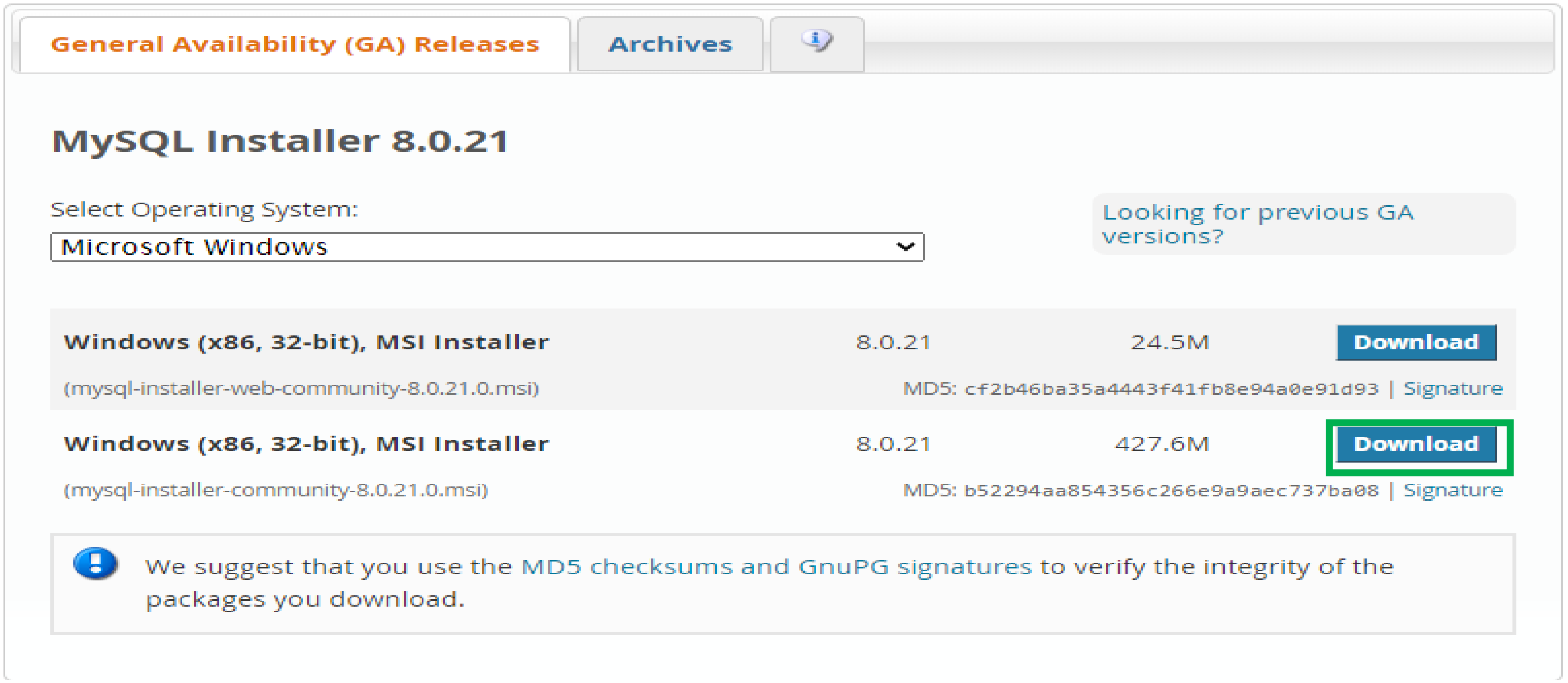
Looking for previous GA versions?

Windows (x86, 32-bit), MSI Installer	8.0.21	24.5M	Download
(mysql-installer-web-community-8.0.21.0.msi)	MD5: cf2b46ba35a4443f41fb8e94a0e91d93 | Signature		
Windows (x86, 32-bit), MSI Installer	8.0.21	427.6M	Download
(mysql-installer-community-8.0.21.0.msi)	MD5: b52294aa854356c266e9a9aec737ba08 | Signature		

We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

MySQL Download

- Official Website - https://www.mysql.com/



MySQL Installer 8.0.21

Select Operating System:
Microsoft Windows

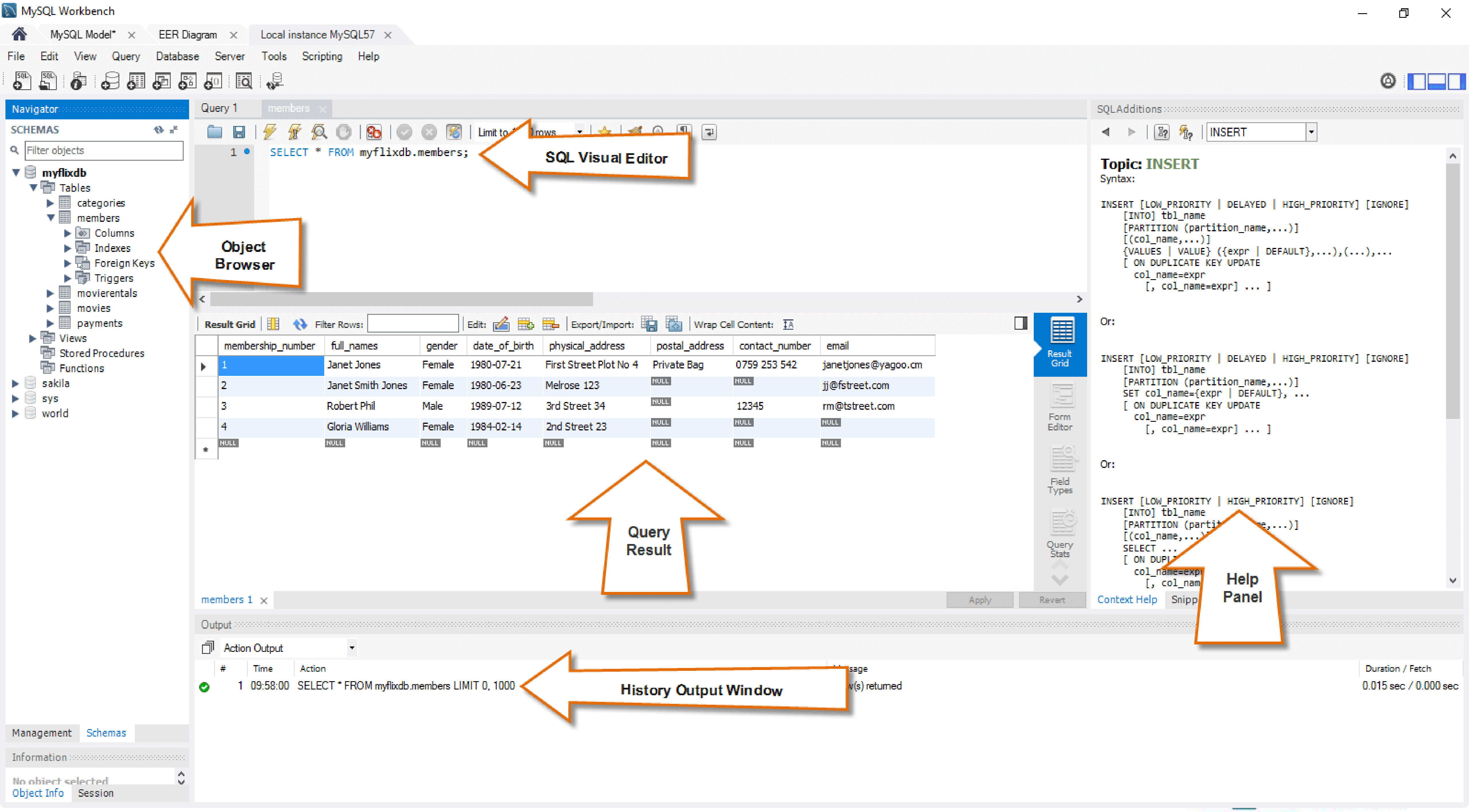
Looking for previous GA versions?

Windows (x86, 32-bit), MSI Installer	8.0.21	24.5M	Download
(mysql-installer-web-community-8.0.21.0.msi)			MD5: cf2b46ba35a4443f41fb8e94a0e91d93 | Signature
Windows (x86, 32-bit), MSI Installer	8.0.21	427.6M	Download
(mysql-installer-community-8.0.21.0.msi)			MD5: b52294aa854356c266e9a9aec737ba08 | Signature

We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

MySQL Workbench

- MySQL Workbench is a unified visual tool for database architects, developers
- Workbench provides
  - data modeling,
  - SQL development,
  - configuration,
  - user administration,
  - backup and much more
- MySQL Workbench is available on Windows, Linux and Mac OS X



```
SELECT * FROM myflixdb.members;
```

membership_number	full_names	gender	date_of_birth	physical_address	postal_address	contact_number	email
1	Janet Jones	Female	1980-07-21	First Street Plot No 4	Private Bag	0759 253 542	janetjones@yagoo.cm
2	Janet Smith Jones	Female	1980-06-23	Melrose 123	NULL	NULL	jj@fstreet.com
3	Robert Phil	Male	1989-07-12	3rd Street 34	NULL	12345	rm@tstreet.com
4	Gloria Williams	Female	1984-02-14	2nd Street 23	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Topic: INSERT
Syntax:

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
    [INTO] tbl_name
    [PARTITION (partition_name,...)]
    [(col_name,...)]
    {VALUES | VALUE} ({expr | DEFAULT},...),(...),...
    [ ON DUPLICATE KEY UPDATE
      col_name=expr
        [, col_name=expr] ... ]
```

Or:

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
    [INTO] tbl_name
    [PARTITION (partition_name,...)]
    SET col_name={expr | DEFAULT}, ...
    [ ON DUPLICATE KEY UPDATE
      col_name=expr
        [, col_name=expr] ... ]
```

Or:

```
INSERT [LOW_PRIORITY | HIGH_PRIORITY] [IGNORE]
    [INTO] tbl_name
    [PARTITION (partit
    [(col_name,...
    SELECT ...
    [ ON DUPL
      col_name=exp
        [, col_nam
```

```
#   Time      Action                                        Message          Duration / Fetch
1   09:58:00  SELECT * FROM myflixdb.members LIMIT 0, 1000  w(s) returned    0.015 sec / 0.000 sec
```

Questions?

Thank You!!!

SQL Basics

Session 2

Agenda

- SQL Commands
- SQL Syntax
- Create Database Objects

SQL Syntax

- Creating a Database
- To create a database, use the CREATE DATEBASE command:
- **MySQL> CREATE DATABASE DATABASE_NAME;**
- **Example CREATE DATABASE SCHOOL_DB;**

SQL Syntax

- Creating a Table
- To create a table, use the CREATE TABLE command:
- **MySQL>**

```
MySQL>   CREATE TABLE STUDENTS_TB
         (STUDENT_ID INT(5),
          STUDENT_NAME VARCHAR(20),
          GENDER CHAR(1));
```

SQL Syntax

- Dropping a Database
- To delete an entire database, use the DROP DATABASE command:
- **MySQL> drop database test_db;**

- Dropping a Table

To delete an entire table, use the DROP TABLE command:

- **MySQL> drop table pet_tb;**

SQL Syntax

Use the INSERT statement to enter data into a table

```
MySQL>    INSERT INTO TABLE VALUES (value1,value2);
          INSERT INTO STUDENTS VALUES (1001,'GURU','M');
```

SQL Syntax

How to view table data-

Use select Command

```
MySQL>    SELECT * FROM TABLE;
           SELECT * FROM STUDENTS;
```

SQL Syntax

Use the UPDATE statement to update data in existing table

MySQL> **UPDATE** table_name **SET** column = value **WHERE** condition

```
UPDATE STUDENTS SET STUDENT_NAME = 'GURU N'
WHERE STUDENT_ID = 1001;
```

DDL - Data Definition Language

Command	Description
CREATE	Creates a new table, a view of a table, or other object in the database
ALTER	Modifies an existing database object, such as a table
DROP	Deletes an entire table, a view of a table or other objects in the database

DML - Data Manipulation Language

Command	Description
INSERT	Creates a record
UPDATE	Modifies record
DELETE	Deletes record

DCL - Data Control Language

Command	Description
GRANT	Gives a privilege to user
REVOKE	Takes back privileges granted from user

DQL - Data Query Language

Command	Description
SELECT	Fetch the data from the database

TCL - Transaction Control Language

Command	Description
COMMIT	Commit command is used to save all the transactions to the database
ROLLBACK	Rollback command is used to undo transactions that have not already been saved
SAVEPOINT	It is used to roll the transaction back to a certain point

SQL Constraints

Session 3

Agenda

- SQL Datatypes
- SQL Constraints

Integer Types (Exact Value) –

INTEGER, INT, SMALLINT, TINYINT, MEDIUMINT, BIGINT

Type	Storage (Bytes)	Minimum Value Signed	Minimum Value Unsigned	Maximum Value Signed	Maximum Value Unsigned
TINYINT	1	-128	0	127	255
SMALLINT	2	-32768	0	32767	65535
MEDIUMINT	3	-8388608	0	8388607	16777215
INT	4	-2147483648	0	2147483647	4294967295
BIGINT	8	-2^{63}	0	$2^{63}-1$	$2^{64}-1$

Fixed-Point Types (Exact Value) – EXACT VALUES

DECIMAL & NUMERIC

- Decimal
- Float
- Double

SQL Constraints

- **NOT NULL** - Ensures that a column cannot have a NULL value
- **UNIQUE** - Ensures that all values in a column are different
- **PRIMARY KEY** - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
- **FOREIGN KEY** - Uniquely identifies a row/record in another table
- **CHECK** - Ensures that all values in a column satisfies a specific condition
- **DEFAULT** - Sets a default value for a column when no value is specified
- **INDEX** - Used to create and retrieve data from the database very quickly

Example of Primary key and Foreign Key

Amazon Example

Example of Primary key and Foreign Key

Amazon Example

Customer Information

Example of Primary key and Foreign Key

Amazon Example

Customer Information

Name	DOR	Address
John	2012-05-11	Bangalore
Suresh	2019-04-12	Mysore
Arjun	2014-12-05	Chennai
Kiran	2012-10-13	Delhi
Kiran	2008-06-18	Calcutta

Example of Primary key and Foreign Key

Amazon Example

Customer Information

Sales Information

Name	DOR	Address
John	2012-05-11	Bangalore
Suresh	2019-04-12	Mysore
Arjun	2014-12-05	Chennai
Kiran	2012-10-13	Delhi
Kiran	2008-06-18	Calcutta

Example of Primary key and Foreign Key

Amazon Example

Customer Information

Name	DOR	Address
John	2012-05-11	Bangalore
Suresh	2019-04-12	Mysore
Arjun	2014-12-05	Chennai
Kiran	2012-10-13	Delhi
Kiran	2008-06-18	Calcutta

Sales Information

Name	Product ID	Sales
John	IPhone	$ 15000
Arjun	Samsung	$ 12500
Arjun	Vivo	$ 9000
Kiran	MI	$ 7500

Example of Primary key and Foreign Key

Amazon Example

Customer Information

Name	DOR	Address
John	2012-05-11	Bangalore
Suresh	2019-04-12	Mysore
Arjun	2014-12-05	Chennai
Kiran	2012-10-13	Delhi
Kiran	2008-06-18	Calcutta

Sales Information

Name	Product ID	Sales
John	IPhone	$ 15000
Arjun	Samsung	$ 12500
Arjun	Vivo	$ 9000
Kiran	MI	$ 7500

Example of Primary key and Foreign Key

Amazon Example

Customer Information

User ID	Name	DOR	Address
X0001	John	2012-05-11	Bangalore
X0002	Suresh	2019-04-12	Mysore
X0003	Arjun	2014-12-05	Chennai
X0004	Kiran	2012-10-13	Delhi
X0005	Kiran	2008-06-18	Calcutta

Sales Information

User ID	Product ID	Sales
X0001	IPhone	$ 15000
X0003	Samsung	$ 12500
X0003	Vivo	$ 9000
X0005	MI	$ 7500

Example of Primary key and Foreign Key

Amazon Example

Customer Information

User ID	Name	DOR	Address
X0001	John	2012-05-11	Bangalore
X0002	Suresh	2019-04-12	Mysore
X0003	Arjun	2014-12-05	Chennai
X0004	Kiran	2012-10-13	Delhi
X0005	Kiran	2008-06-18	Calcutta

Sales Information

User ID	Product ID	Sales
X0001	IPhone	$ 15000
X0003	Samsung	$ 12500
X0003	Vivo	$ 9000
X0005	MI	$ 7500

Example of Primary key and Foreign Key

Amazon Example

Customer Information

User ID	Name	DOR	Address
X0001	John	2012-05-11	Bangalore
X0002	Suresh	2019-04-12	Mysore
X0003	Arjun	2014-12-05	Chennai
X0004	Kiran	2012-10-13	Delhi
X0005	Kiran	2008-06-18	Calcutta

Sales Information

User ID	Product ID	Sales
X0001	IPhone	$ 15000
X0003	Samsung	$ 12500
X0003	Vivo	$ 9000
X0005	MI	$ 7500

Primary Key

Example of Primary key and Foreign Key

Amazon Example

Customer Information

User ID	Name	DOR	Address
X0001	John	2012-05-11	Bangalore
X0002	Suresh	2019-04-12	Mysore
X0003	Arjun	2014-12-05	Chennai
X0004	Kiran	2012-10-13	Delhi
X0005	Kiran	2008-06-18	Calcutta

Sales Information

User ID	Product ID	Sales
X0001	IPhone	$ 15000
X0003	Samsung	$ 12500
X0003	Vivo	$ 9000
X0005	MI	$ 7500

Primary Key

Example of Primary key and Foreign Key

Amazon Example

Customer Information

User ID	Name	DOR	Address
X0001	John	2012-05-11	Bangalore
X0002	Suresh	2019-04-12	Mysore
X0003	Arjun	2014-12-05	Chennai
X0004	Kiran	2012-10-13	Delhi
X0005	Kiran	2008-06-18	Calcutta

Primary Key

Sales Information

User ID	Product ID	Sales
X0001	IPhone	$ 15000
X0003	Samsung	$ 12500
X0003	Vivo	$ 9000
X0005	MI	$ 7500

Foreign Key

SQL Joins

Session 4

Agenda

- SQL Joins

SQL Joins

- Inner join
- Left join
- Right join
- Full join

Example of Primary key and Foreign Key

Amazon Example

Example of Primary key and Foreign Key

Amazon Example

Customer Information

Example of Primary key and Foreign Key

Amazon Example

Customer Information

Name	DOR	Address
John	2012-05-11	Bangalore
Suresh	2019-04-12	Mysore
Arjun	2014-12-05	Chennai
Kiran	2012-10-13	Delhi
Kiran	2008-06-18	Calcutta

Example of Primary key and Foreign Key

Amazon Example

Customer Information

Sales Information

Name	DOR	Address
John	2012-05-11	Bangalore
Suresh	2019-04-12	Mysore
Arjun	2014-12-05	Chennai
Kiran	2012-10-13	Delhi
Kiran	2008-06-18	Calcutta

Example of Primary key and Foreign Key

Amazon Example

Customer Information

Name	DOR	Address
John	2012-05-11	Bangalore
Suresh	2019-04-12	Mysore
Arjun	2014-12-05	Chennai
Kiran	2012-10-13	Delhi
Kiran	2008-06-18	Calcutta

Sales Information

Name	Product ID	Sales
John	IPhone	$ 15000
Arjun	Samsung	$ 12500
Arjun	Vivo	$ 9000
Kiran	MI	$ 7500

Example of Primary key and Foreign Key

Amazon Example

Customer Information

Name	DOR	Address
John	2012-05-11	Bangalore
Suresh	2019-04-12	Mysore
Arjun	2014-12-05	Chennai
Kiran	2012-10-13	Delhi
Kiran	2008-06-18	Calcutta

Sales Information

Name	Product ID	Sales
John	IPhone	$ 15000
Arjun	Samsung	$ 12500
Arjun	Vivo	$ 9000
Kiran	MI	$ 7500

Example of Primary key and Foreign Key

Amazon Example

Customer Information

User ID	Name	DOR	Address
X0001	John	2012-05-11	Bangalore
X0002	Suresh	2019-04-12	Mysore
X0003	Arjun	2014-12-05	Chennai
X0004	Kiran	2012-10-13	Delhi
X0005	Kiran	2008-06-18	Calcutta

Sales Information

User ID	Product ID	Sales
X0001	IPhone	$ 15000
X0003	Samsung	$ 12500
X0003	Vivo	$ 9000
X0005	MI	$ 7500

Example of Primary key and Foreign Key

Amazon Example

Customer Information

User ID	Name	DOR	Address
X0001	John	2012-05-11	Bangalore
X0002	Suresh	2019-04-12	Mysore
X0003	Arjun	2014-12-05	Chennai
X0004	Kiran	2012-10-13	Delhi
X0005	Kiran	2008-06-18	Calcutta

Sales Information

User ID	Product ID	Sales
X0001	IPhone	$ 15000
X0003	Samsung	$ 12500
X0003	Vivo	$ 9000
X0005	MI	$ 7500

Example of Primary key and Foreign Key

Amazon Example

Customer Information

User ID	Name	DOR	Address
X0001	John	2012-05-11	Bangalore
X0002	Suresh	2019-04-12	Mysore
X0003	Arjun	2014-12-05	Chennai
X0004	Kiran	2012-10-13	Delhi
X0005	Kiran	2008-06-18	Calcutta

Sales Information

User ID	Product ID	Sales
X0001	IPhone	$ 15000
X0003	Samsung	$ 12500
X0003	Vivo	$ 9000
X0005	MI	$ 7500

Primary Key

Example of Primary key and Foreign Key

Amazon Example

Customer Information

User ID	Name	DOR	Address
X0001	John	2012-05-11	Bangalore
X0002	Suresh	2019-04-12	Mysore
X0003	Arjun	2014-12-05	Chennai
X0004	Kiran	2012-10-13	Delhi
X0005	Kiran	2008-06-18	Calcutta

Sales Information

User ID	Product ID	Sales
X0001	IPhone	$ 15000
X0003	Samsung	$ 12500
X0003	Vivo	$ 9000
X0005	MI	$ 7500

Primary Key

Example of Primary key and Foreign Key

Amazon Example

Customer Information

User ID	Name	DOR	Address
X0001	John	2012-05-11	Bangalore
X0002	Suresh	2019-04-12	Mysore
X0003	Arjun	2014-12-05	Chennai
X0004	Kiran	2012-10-13	Delhi
X0005	Kiran	2008-06-18	Calcutta

Sales Information

User ID	Product ID	Sales
X0001	IPhone	$ 15000
X0003	Samsung	$ 12500
X0003	Vivo	$ 9000
X0005	MI	$ 7500

Primary Key

Foreign Key

JOINS

- Commands which are used to combine rows from two or more tables
- In real world, not all data will be present in 1 table
- For reporting purpose, you need complete information
- Data understandability

Example

- Customer Information

ID	Name	Age	Address	Wallet Bal
1	Ramesh	32	Ahmedabad	200
2	Khilan	23	Delhi	7500
3	kaushik	25	Kota	12000
4	Chaitali	22	Mumbai	6500
5	Hardik	27	Bhopal	600
6	Komal	26	MP	750
7	Muffy	27	Indore	500

Example

- Customer Information

ID	Name	Age	Address	Wallet Bal
1	Ramesh	32	Ahmedabad	200
2	Khilan	23	Delhi	7500
3	kaushik	25	Kota	12000
4	Chaitali	22	Mumbai	6500
5	Hardik	27	Bhopal	600
6	Komal	26	MP	750
7	Muffy	27	Indore	500

Order Information

OID	Date	ID	Amount
102	2009-10-08	3	3000
100	2009-10-08	3	1500
101	2009-11-20	2	1560
103	2008-05-20	4	2060

Example

- Which Customers did not place any order?

ID	Name	Age	Address	Wallet Bal
1	Ramesh	32	Ahmedabad	200
2	Khilan	23	Delhi	7500
3	kaushik	25	Kota	12000
4	Chaitali	22	Mumbai	6500
5	Hardik	27	Bhopal	600
6	Komal	26	MP	750
7	Muffy	27	Indore	500

OID	Date	C_ID	Amount
102	2009-10-08	3	3000
100	2009-10-08	3	1500
101	2009-11-20	2	1560
103	2008-05-20	4	2060

Example

- Who is my loyal Customer?

ID	Name	Age	Address	Wallet Bal
1	Ramesh	32	Ahmedabad	200
2	Khilan	23	Delhi	7500
3	kaushik	25	Kota	12000
4	Chaitali	22	Mumbai	6500
5	Hardik	27	Bhopal	600
6	Komal	26	MP	750
7	Muffy	27	Indore	500

OID	Date	C_ID	Amount
102	2009-10-08	3	3000
100	2009-10-08	3	1500
101	2009-11-20	2	1560
103	2008-05-20	4	2060

Inner Join

- Records should match on both tables

ID	Name	Age	Address	Wallet Bal
1	Ramesh	32	Ahmedabad	200
2	Khilan	23	Delhi	7500
3	kaushik	25	Kota	12000
4	Chaitali	22	Mumbai	6500
5	Hardik	27	Bhopal	600
6	Komal	26	MP	750
7	Muffy	27	Indore	500

OID	Date	C_ID	Amount
102	2009-10-08	3	3000
100	2009-10-08	3	1500
101	2009-11-20	2	1560
103	2008-05-20	4	2060

Inner Join

- Records should match on both tables

ID	Name	Age	Address	Wallet Bal
1	Ramesh	32	Ahmedabad	200
2	Khilan	23	Delhi	7500
3	kaushik	25	Kota	12000
4	Chaitali	22	Mumbai	6500
5	Hardik	27	Bhopal	600
6	Komal	26	MP	750
7	Muffy	27	Indore	500

OID	Date	C_ID	Amount
102	2009-10-08	3	3000
100	2009-10-08	3	1500
101	2009-11-20	2	1560
103	2008-05-20	4	2060

Lets say, I need details of Customer Name, address, what order he place, when he placed?

Inner Join

- Records should match on both tables

ID	Name	Age	Address	Wallet Bal
1	Ramesh	32	Ahmedabad	200
2	Khilan	23	Delhi	7500
3	kaushik	25	Kota	12000
4	Chaitali	22	Mumbai	6500
5	Hardik	27	Bhopal	600
6	Komal	26	MP	750
7	Muffy	27	Indore	500

OID	Date	C_ID	Amount
102	2009-10-08	3	3000
100	2009-10-08	3	1500
101	2009-11-20	2	1560
103	2008-05-20	4	2060

Lets say, I need details of Customer Name, address, what order he place, when he placed?

Inner Join

- Records should match on both tables

ID	Name	Age	Address	Wallet Bal
1	Ramesh	32	Ahmedabad	200
2	Khilan	23	Delhi	7500
3	kaushik	25	Kota	12000
4	Chaitali	22	Mumbai	6500
5	Hardik	27	Bhopal	600
6	Komal	26	MP	750
7	Muffy	27	Indore	500

OID	Date	C_ID	Amount
102	2009-10-08	3	3000
100	2009-10-08	3	1500
101	2009-11-20	2	1560
103	2008-05-20	4	2060

NAME	Address	OID	Date
Khilan	Delhi	101	2009-11-20
kaushik	Kota	102	2009-10-08
kaushik	Kota	100	2009-10-08
Chaitali	Mumbai	103	2008-05-20

Lets say, I need details of Customer Name, address, what order he place, when he placed?

Left Join

- Display all records from A table

ID	Name	Age	Address	Wallet Bal
1	Ramesh	32	Ahmedabad	200
2	Khilan	23	Delhi	7500
3	kaushik	25	Kota	12000
4	Chaitali	22	Mumbai	6500
5	Hardik	27	Bhopal	600
6	Komal	26	MP	750
7	Muffy	27	Indore	500

Left Join

- Display all records from A table

OID	Date	C_ID	Amount
102	2009-10-08	3	3000
100	2009-10-08	3	1500
101	2009-11-20	2	1560
103	2008-05-20	4	2060

ID	Name	Age	Address	Wallet Bal
1	Ramesh	32	Ahmedabad	200
2	Khilan	23	Delhi	7500
3	kaushik	25	Kota	12000
4	Chaitali	22	Mumbai	6500
5	Hardik	27	Bhopal	600
6	Komal	26	MP	750
7	Muffy	27	Indore	500

Left Join

- Display all records from A table

ID	Name	Age	Address	Wallet Bal
1	Ramesh	32	Ahmedabad	200
2	Khilan	23	Delhi	7500
3	kaushik	25	Kota	12000
4	Chaitali	22	Mumbai	6500
5	Hardik	27	Bhopal	600
6	Komal	26	MP	750
7	Muffy	27	Indore	500

OID	Date	C_ID	Amount
102	2009-10-08	3	3000
100	2009-10-08	3	1500
101	2009-11-20	2	1560
103	2008-05-20	4	2060

NAME	Address	OID	Date
Khilan	Delhi	101	2009-11-20
kaushik	Kota	102	2009-10-08
kaushik	Kota	100	2009-10-08
Chaitali	Mumbai	103	2008-05-20
Ramesh	Ahmedabad	Null	Null
Hardik	Bhopal	Null	Null
Komal	MP	Null	Null
Muffy	Indore	Null	Null

Right Join

- Display all records from B table

ID	Name	Age	Address	Wallet Bal
1	Ramesh	32	Ahmedabad	200
2	Khilan	23	Delhi	7500
3	kaushik	25	Kota	12000
4	Chaitali	22	Mumbai	6500
5	Hardik	27	Bhopal	600
6	Komal	26	MP	750
7	Muffy	27	Indore	500

Right Join

- Display all records from B table

ID	Name	Age	Address	Wallet Bal
1	Ramesh	32	Ahmedabad	200
2	Khilan	23	Delhi	7500
3	kaushik	25	Kota	12000
4	Chaitali	22	Mumbai	6500
5	Hardik	27	Bhopal	600
6	Komal	26	MP	750
7	Muffy	27	Indore	500

OID	Date	C_ID	Amount
102	2009-10-08	3	3000
100	2009-10-08	3	1500
101	2009-11-20	2	1560
103	2008-05-20	4	2060
199	2010-12-31	-	9000

Right Join

- Display all records from B table

ID	Name	Age	Address	Wallet Bal
1	Ramesh	32	Ahmedabad	200
2	Khilan	23	Delhi	7500
3	kaushik	25	Kota	12000
4	Chaitali	22	Mumbai	6500
5	Hardik	27	Bhopal	600
6	Komal	26	MP	750
7	Muffy	27	Indore	500

OID	Date	C_ID	Amount
102	2009-10-08	3	3000
100	2009-10-08	3	1500
101	2009-11-20	2	1560
103	2008-05-20	4	2060
199	2010-12-31	-	9000

Right Join

- Display all records from B table

ID	Name	Age	Address	Wallet Bal
1	Ramesh	32	Ahmedabad	200
2	Khilan	23	Delhi	7500
3	kaushik	25	Kota	12000
4	Chaitali	22	Mumbai	6500
5	Hardik	27	Bhopal	600
6	Komal	26	MP	750
7	Muffy	27	Indore	500

OID	Date	C_ID	Amount
102	2009-10-08	3	3000
100	2009-10-08	3	1500
101	2009-11-20	2	1560
103	2008-05-20	4	2060
199	2010-12-31	-	9000

NAME	Address	OID	Date
Khilan	Delhi	101	2009-11-20
kaushik	Kota	102	2009-10-08
kaushik	Kota	100	2009-10-08
Chaitali	Mumbai	103	2008-05-20
Null	Null	199	2010-12-31

Full Join

- Display all records from A table

ID	Name	Age	Address	Wallet Bal
1	Ramesh	32	Ahmedabad	200
2	Khilan	23	Delhi	7500
3	kaushik	25	Kota	12000
4	Chaitali	22	Mumbai	6500
5	Hardik	27	Bhopal	600
6	Komal	26	MP	750
7	Muffy	27	Indore	500

Full Join

- Display all records from A & B table

ID	Name	Age	Address	Wallet Bal
1	Ramesh	32	Ahmedabad	200
2	Khilan	23	Delhi	7500
3	kaushik	25	Kota	12000
4	Chaitali	22	Mumbai	6500
5	Hardik	27	Bhopal	600
6	Komal	26	MP	750
7	Muffy	27	Indore	500

OID	Date	C_ID	Amount
102	2009-10-08	3	3000
100	2009-10-08	3	1500
101	2009-11-20	2	1560
103	2008-05-20	4	2060
199	2008-05-20	-	1253

Full Join

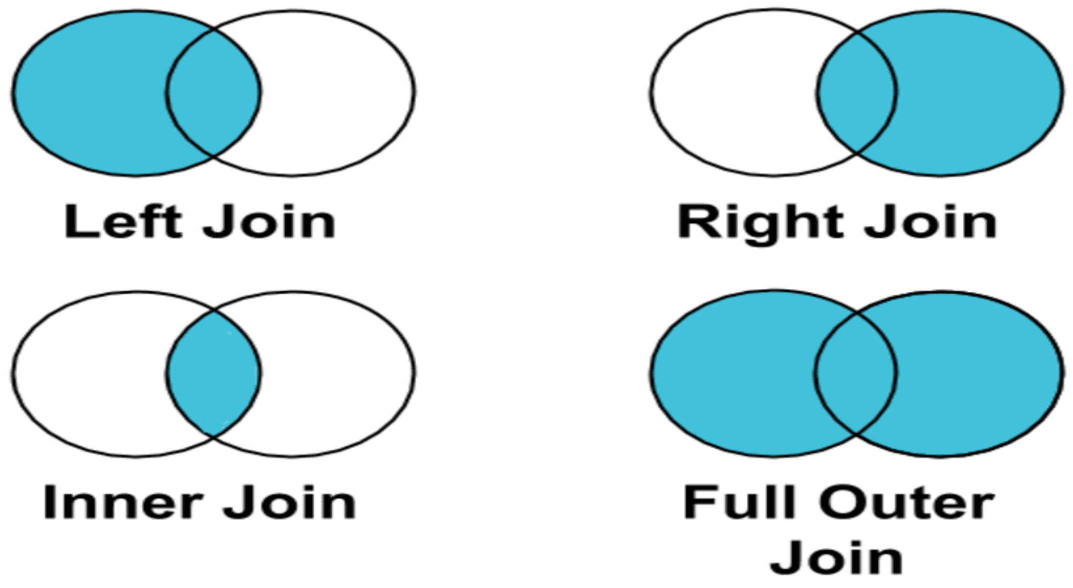
- Display all records from A, B table

ID	Name	Age	Address	Wallet Bal
1	Ramesh	32	Ahmedabad	200
2	Khilan	23	Delhi	7500
3	kaushik	25	Kota	12000
4	Chaitali	22	Mumbai	6500
5	Hardik	27	Bhopal	600
6	Komal	26	MP	750
7	Muffy	27	Indore	500

OID	Date	C_ID	Amount
102	2009-10-08	3	3000
100	2009-10-08	3	1500
101	2009-11-20	2	1560
103	2008-05-20	4	2060
199	2008-05-20	-	1253

NAME	Address	OID	Date
Khilan	Delhi	101	2009-11-20
kaushik	Kota	102	2009-10-08
kaushik	Kota	100	2009-10-08
Chaitali	Mumbai	103	2008-05-20
Ramesh	Ahmedabad	Null	Null
Hardik	Bhopal	Null	Null
Komal	MP	Null	Null
Muffy	Indore	Null	Null
Null	Null	199	2008-05-20

SQL Joins



PL / SQL

Agenda

- SQL vs. PLSQL
- Procedures
- Functions
- Views
- Triggers

PL/SQL

- PL/SQL is a combination of SQL along with the procedural features of programming languages
- PL/SQL adds many procedural constructs to SQL language to overcome some limitations of SQL
- **SQL** executes the single query at a time whereas, **PL/SQL** executes the block of code at once

BASIS FOR COMPARISON	SQL	PL/SQL
Basic	In SQL you can execute a single query or a command at a time.	In PL/SQL you can execute a block of code at a time.
Full form	Structured Query Language	Procedural Language, extension of SQL.
Purpose	It is like a source of data that is to be displayed.	It is language that creates an application that display's the data acquired by SQL.
Writes	In SQL you can write queries and command using DDL, DML statements.	In PL/SQL you can write block of code that has procedures, functions, packages or variables, etc.
Use	Using SQL, you can retrieve, modify, add, delete, or manipulate the data in the database.	Using PL/SQL, you can create applications or server pages that display's the information obtained from SQL in a proper format.

Procedures

- Procedure is a stored program that you can pass parameters into and get results as you desire
- A **procedure** is a subroutine (like a subprogram)
- Procedure has a name, a parameter list, and SQL statement(s)

Procedures

- Syntax

```
DELIMITER $$
CREATE PROCEDURE GetAllProducts()
    BEGIN
        SELECT * FROM products;
    END
$$ DELIMITER ;
```

Functions

- As the name implies, you can perform any logic
- You can write any mathematical operations on data
- A function is same as a procedure except that it returns a value

Functions

- Syntax

```
DELIMITER //
CREATE FUNCTION Sum_Ab (a int, b int)
    RETURNS int
          Begin
                    return a + b;
          End;//
 DELIMITER ;
```

Views

- **VIEWS** are virtual tables that do not store any data of their own but display data stored in other tables
- In other words, VIEWS are nothing but SQL Queries
- MySQL view can show data from one table or many tables.

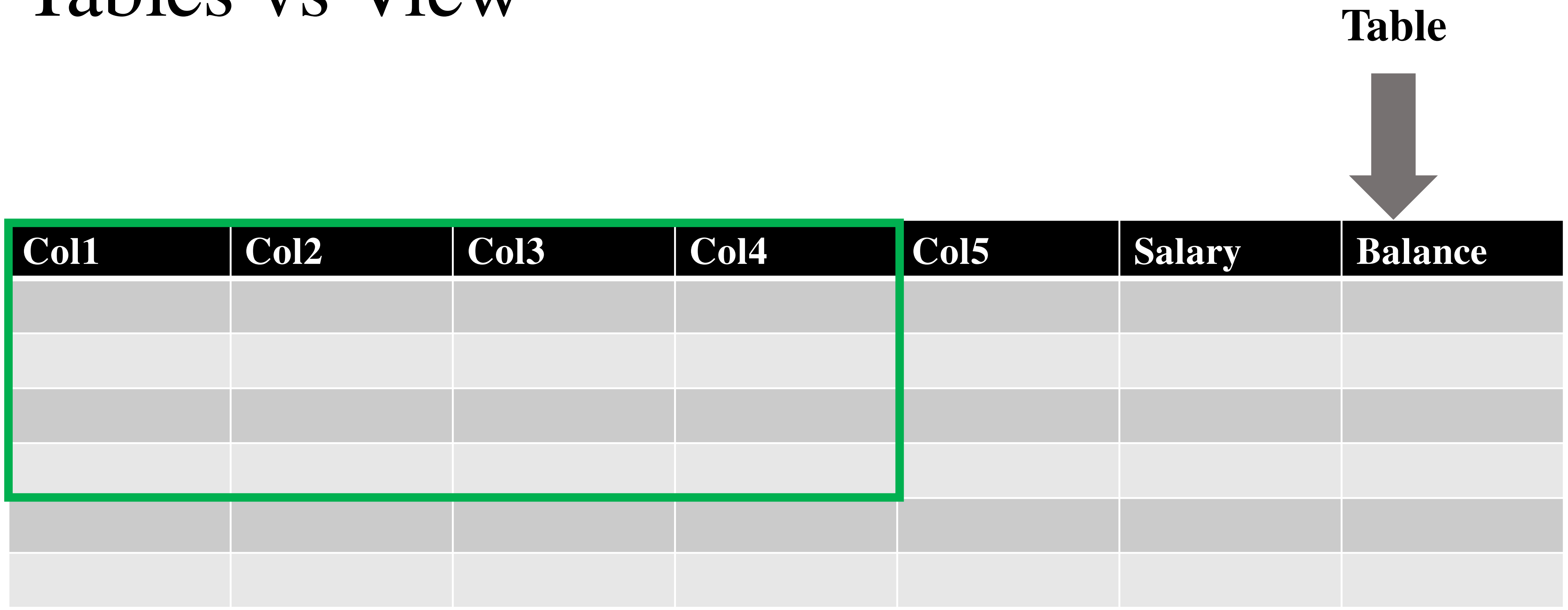
Tables vs View

Col1	Col2	Col3	Col4	Col5	Salary	Balance

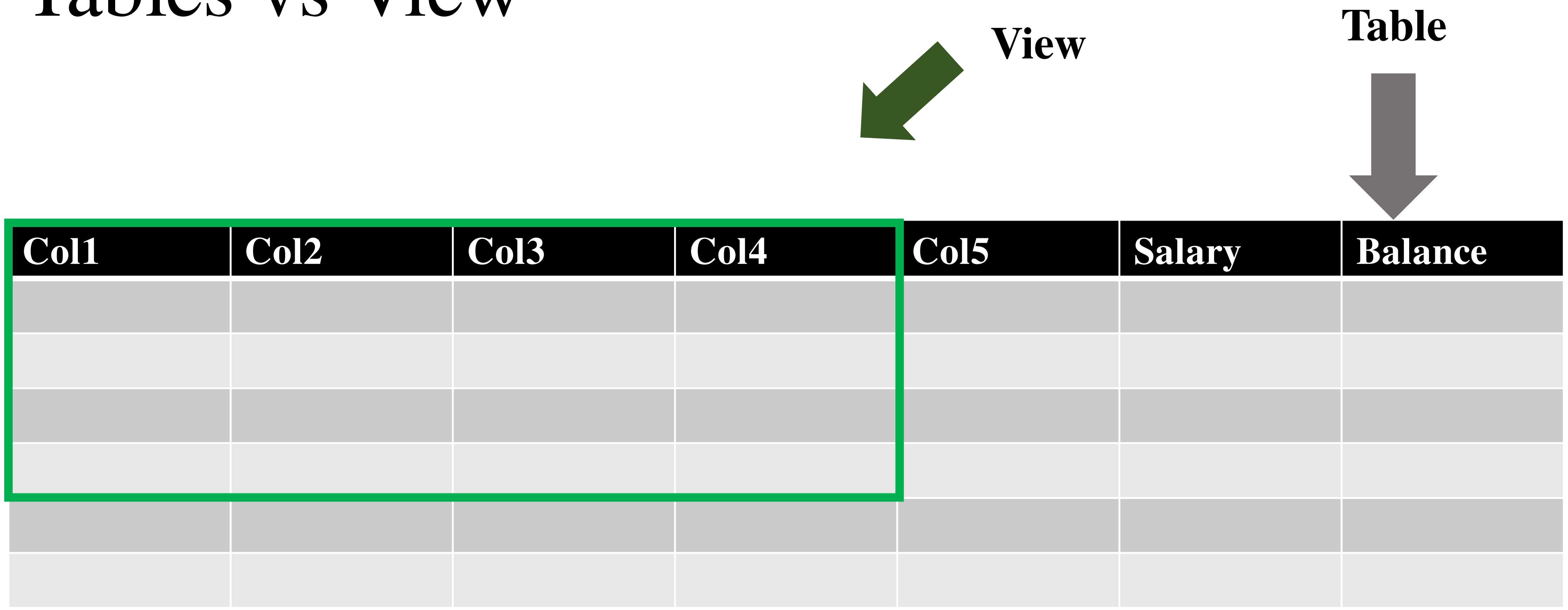
Tables vs View

Col1	Col2	Col3	Col4	Col5	Salary	Balance

Tables vs View

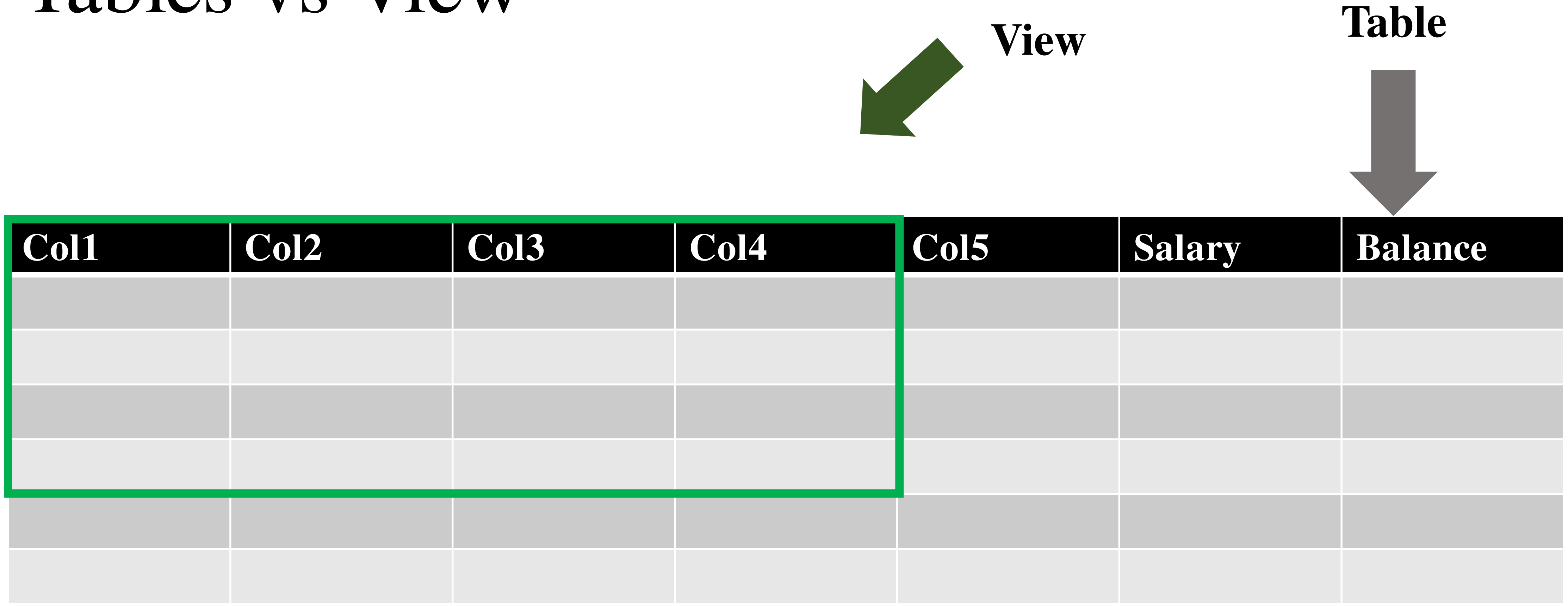


Tables vs View



Col1	Col2	Col3	Col4	Col5	Salary	Balance

Tables vs View



Col1	Col2	Col3	Col4	Col5	Salary	Balance

Views

- Customer Information

Cust ID	Name	Address
1	Santhosh	Bangalore
2	Kiran	Chennai
3	Mahdi	Mumbai
4	Taren	Bangalore
5	Dinesh	Chennai
6	Varun	Bangalore

Views

- Customer Information

Cust ID	Name	Address
1	Santhosh	Bangalore
2	Kiran	Chennai
3	Mahdi	Mumbai
4	Taren	Bangalore
5	Dinesh	Chennai
6	Varun	Bangalore

*You can go for **GRANTS** to give permission to a table. But here the scenario is different*

Views

- Customer Information

Cust ID	Name	Address
1	Santhosh	Bangalore
2	Kiran	Chennai
3	Mahdi	Mumbai
4	Taren	Bangalore
5	Dinesh	Chennai
6	Varun	Bangalore

Sales Information

Cust ID	Product	Sales
1	Vivo	9500
2	IPhone	25000
4	Redmi	7500
6	Samsung	8500
5	Samsung	6500

Views

- Customer Information

Sales Information

Cust ID	Name	Address
1	Santhosh	Bangalore
2	Kiran	Chennai
3	Mahdi	Mumbai
4	Taren	Bangalore
5	Dinesh	Chennai
6	Varun	Bangalore

Cust ID	Product	Sales
1	Vivo	9500
2	IPhone	25000
4	Redmi	7500
6	Samsung	8500
5	Samsung	6500

Views

- Customer Information

Cust ID	Name	Address
1	Santhosh	Bangalore
2	Kiran	Chennai
3	Mahdi	Mumbai
4	Taren	Bangalore
5	Dinesh	Chennai
6	Varun	Bangalore

Sales Information

Cust ID	Product	Sales
1	Vivo	9500
2	IPhone	25000
4	Redmi	7500
6	Samsung	8500
5	Samsung	6500

Views

- Customer Information

Cust ID	Name	Address
1	Santhosh	Bangalore
2	Kiran	Chennai
3	Mahdi	Mumbai
4	Taren	Bangalore
5	Dinesh	Chennai
6	Varun	Bangalore

Sales Information

Cust ID	Product	Sales
1	Vivo	9500
2	IPhone	25000
4	Redmi	7500
6	Samsung	8500
5	Samsung	6500

Views

- Customer Information

Cust ID	Name	Address
1	Santhosh	Bangalore
2	Kiran	Chennai
3	Mahdi	Mumbai
4	Taren	Bangalore
5	Dinesh	Chennai
6	Varun	Bangalore

Sales Information

Cust ID	Product	Sales
1	Vivo	9500
2	IPhone	25000
4	Redmi	7500
6	Samsung	8500
5	Samsung	6500

Views

- Customer Information

Cust ID	Name	Address
1	Santhosh	Bangalore
2	Kiran	Chennai
3	Mahdi	Mumbai
4	Taren	Bangalore
5	Dinesh	Chennai
6	Varun	Bangalore

Sales Information

Cust ID	Product	Sales
1	Vivo	9500
2	IPhone	25000
4	Redmi	7500
6	Samsung	8500
5	Samsung	6500

Views

- Customer Information

Cust ID	Name	Address
1	Santhosh	Bangalore
2	Kiran	Chennai
3	Mahdi	Mumbai
4	Taren	Bangalore
5	Dinesh	Chennai
6	Varun	Bangalore

Sales Information

Cust ID	Product	Sales
1	Vivo	9500
2	IPhone	25000
4	Redmi	7500
6	Samsung	8500
5	Samsung	6500

Views

- Customer Information

Sales Information

Cust ID	Name	Address
1	Santhosh	Bangalore
2	Kiran	Chennai
3	Mahdi	Mumbai
4	Taren	Bangalore
5	Dinesh	Chennai
6	Varun	Bangalore

Cust ID	Product	Sales
1	Vivo	9500
2	IPhone	25000
4	Redmi	7500
6	Samsung	8500
5	Samsung	6500

Name	Address	Product	Sales

Views

Name	Address	Product	Sales
Santhosh	Bangalore	Vivo	9500
Taren	Bangalore	Redmi	7500
Varun	Bangalore	Samsung	8500

```
SELECT A.NAME, A.ADDRESS, B.PRODUCT, B.SALES FROM CUSTUMER A LEFT JOIN PRO_SALES B
ON A.ID = B.ID
WHERE A.ADDRESS= 'BANGALORE';
```

Triggers

- In MySQL, a trigger is a stored program invoked automatically in response to an event such as insert, update, or delete that occurs in the associated table
- Triggers can be useful for auditing the data changes in tables
- Triggers give an alternative way to run scheduled tasks

Recap of SQL Course

- Need of SQL
- OLAP vs OLTP
- SQL databases available in market
- MYSQL
- Installations
- SQL Commands, Syntax
- SQL Objects
- SQL DDL, DML, DCL, TCL

Recap of SQL Course Conti…

- SQL Constraints
- SQL data retrieval (where, between, order by, group by, having Clause)
- SQL Joins
- SQL Advanced
- PLSQL
- Procedures, Functions & Views
- Import and Export data