

MY SQL

Udemy.co

Date: _____

- SQL vs MySQL, Database ANSI
- Create Database, Get data
- Complex queries (Not case sensitive)
- Date Times, String funⁿ, Log operation

[*] Sequel → Manipulate & Access database

Entities → fields → Cust Id, Cust Name, Address
(Top Row), Vertical

Records → Row → Individual Entry → Horizontal

Selects → Extracts data, Distinct

Select * from Customers

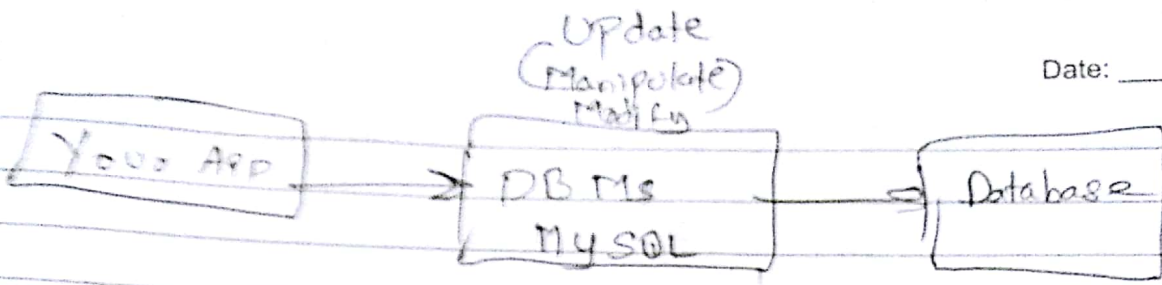
Where Country = 'Germany' AND City = 'Berlin';

Order by Asc, Desc

* Database ⇒ Collection of Data → Rows & Columns

→ Method for Accessing & manipulating data.

Database vs DBMS or ROBMS ⇒ Collection of data
Different → having interface for that data



* Database

→ Structured set of Computerized data with an accessible interface.

* MySQL VS SQL Sequel

~~IBM~~ DBMS, Oracle, → use sql

SQL → Lang → interact & access the data.
& manipulate

↳ "Talk" to our database. → Queries

SQL ⇒ Standard Closed - CK6 TV
signal

Cloud 9 ⇒ Online Interactive development Environment

↳ Text Editor

↳ Cloud based developer Environment

⇒ Workspace ⇒ Cloud 9 has a server ^{pc} somewhere.

File tree ⇒ contents of our workspace

middle area ⇒ Text Editor ⇒ Bash ^{the} cell

MySQL on Cloud ?

★ Service 1

Date: _____

→ RDBMS → Multiple user access to a no. of database

MySQL-ctl (start)
(stop) *

CLI

→ Command line Interface
→ Interactive shell

CTL → File Extension → Control
source code for visual basic

\q; , exit; , quit; (ctrl-c)

help; , show databases;

select @hostname;

SECTION-3 ⇒ Creating Databases

Create database DogAPP;

Drop database <name>;

⇒ Create table pastries

C

name varchar(50)

quantity (INT);

Create DATABASE, Show database;
 Use < database name >;
 Select database ();
 Drop database

Use C9;

NULL → (SELECT)

* Introduction to tables

Database → bunch of table.
 Relational

Tables → Collection of data in a structured format.
 (Holds the data) related

* Data types

Numeric

String

Int Smallint
 Tinyint, mediumint
 Bigint, Decimal
 Numeric, Float
 Double, BIT

⇒ Char, VarChar,
 Binary, Var Binary

whole no.

→ 400 Euros

⇒ Text/strings

Variable-lengths

Text ⇒ Varchar(100)

Between 1 & 255 character

⇒ Username
↳ Varchar (15)

Tweet Content
Varchar (140)

No. of Favourites
Date: 5
Int

* Create table (tablename)

```
(  
  Column_name data-type,  
  Column_name data-type  
);
```

Create table cats

```
(  
  name Varchar(100),  
  age int  
);
```

} ⇒ show tables;

Show tables;

Show Columns from <table name>;
(Describe)

↳ DESC <tablename>;

Delete + Drop TABLE cats;

< Inserting Data

← Name of the table

Insert into cats (name, age)
Values ("Jatson", 7);

SELECT, Filtering & Sorting

⇒ Select * from cats;

* Multiple Insert ⇒ Values

* Warnings

* Null ⇒ Not mentioned } Not known

↳ Not known

→ Not Null

Not null → Empty string

* Default values

↳ How to specify Default 'unnamed',

PRIMARY KEY ⇒ Uniquely referred.

* Id ↳ Unique identifier

CRUD

C - Create

Select &

R - Read

Insert

U - Update

Update & D

~~X~~ - Delete

* Create

Date: _____

→ Insert into

* Searching or Reading Data out

↳ Retrieve

↳ SELECT

* Where

* ALIASES

→ Easier to read results.

select Cat_id, AS id, name from cats;

* UPDATE → Update set
↳ where.

Update Cats Set age = 12

Where breed = 'Maine Coon';

* DELETE

→ Delete from Cats
Where name 'Egg';

CRUD

Date: _____

Super Ultra Hyper Mega Exercise

Use <database name>

* Running SQL Files

⇒ String Functions

* Reading Data ⇒ select

first-file-sql

RUN ⇒ source file-name-sql
↓
book-data.sql

source
testing/insert.sql

Loading our Book data

⇒ MySQL String functions

* Working with CONCAT

CONCAT → Combine Data for cleaner O/P.

CONCAT (X, Y, Z) → (fname, last+name)
(Column, another column)

Date: _____

CONCAT (author-fname, ' ', author-lastname)

select CONCAT ('Hello', 'world');
from Books;

Printing
o/p

⇒ Database does not change.

~~SQL~~

select author-fname as first;
author-lastname as last

* CONCAT_WS

↳ with separator (-),

↳ Add space

* Substring → select individual parts of strings

select substring ('Hello world', 1, 4)

↳ starts from 1 → HELLO

7
↳ world

Replace

→ Replace parts of strings.

REVERSE

→ select Reverse ('author_name') from books;

* CHAR_LENGTH

→ Counts characters in string.

⇒ select char_length('Hello world');
 'is', 'Characters long) concat'

* UPPER & LOWER() { Changing case }

⇒ select upper ('Hello world');

* USING DISTINCT

* Order By → Sorting our results
 ↳ Alphabetical order.

* LIMIT → Starts from 0

Date: _____

- 1st two best selling books
- Top 10 most recent books

Order by Sorting

Select title, released_year from books
Order by released_year Desc limit 5,

LIKE

→ Better searching

Where author_name

Like ' %da%'

Starts with → 'da%'

↑ ↑
wild cards.
%/%/%

'_____' (Underscore) (no. search)

↳ 4 characters long (4 digits)

Magic of Aggregate function

⇒ Finding averages
Bunch of data together → Grouping things

Pg no. per author, quantities

* Count function → selecting the data on
→ sorting → order by, limit

* Aggregate fun → Avg page length → for each author

⇒ select * from books

⇒ select ^{distinct} count(*) from books;

select count(distinct 'Author-name') from books;

* Group by → case of aggregating data.

Summarizes or aggregates identical data into single rows.

Count (*) from books, Group by author_name

Min & Max recent year Date _____

↳ Select min(released-year) from books;
↳ 1st book released

Select max(pages)
from books;

select * from books
where pages = (select min(pages)
from books);

Min & Max

Speed Slow
↳ 2 queries

select * from books
order by pages asc limit 1

} same

* Min/Max with Group By { fname & last name }
↳ last name, fname

* SUM → Add things together

Sum all pages in the entire database.

* Sum + Group By

* Average ⇒ AVG

Select avg(released-year) from books;
four decimals

Select (count(*), Released Year,
Avg Stk quantity

Date: _____

Group by Released year

* Data Types → Data, Day, Currency, Storing Decimal points.
How to store them?

Char
↳ Fixed length ⇒ faster for fixed char length.

* Numbers ⇒ INT → Whole nos.

Numeric Data types

① Decimal

Decimal (5,2)

Total no. of digits

Digits after decimal

Fixed-point type

Cat-Excat

999.99

→ Decimal (5,2)

* Float & Double → Floating-point types
(7) (14)

* Dates & Times = DATE → (no time)

* TIME ⇒ DATETIME

* CURDATE, CURTIME & NOW

Date: _____

%i (13 min-50sec)

* Formatting Dates

Day(), Dayname(), Day of week(), Day of Year()

Select name, Day(birthdate) from people;
minute(birthtime)

* DATEDIFF (Now(), birthdate)
(a - b)

→ Interval 3 quarter

(meta)
= TIMESTAMPS

VS DATETIME → (2x) of ^{SBR} Timestamp

(Created_at Timestamp Default now() → Current time
changed ...)

↳ on update current_timestamp

Smaller Range

* Logical Operators

Not Equal \rightarrow $!=$

* Not like

\rightarrow Match patterns in strings

* Greater than

* less than $=$

* Logical AND $==$

\rightarrow \wedge OR

* Between

* Between \cdot x and y

* In & Not In \rightarrow Set of values

\rightarrow Modulo $=$ remainder

* Case statements else Then, End ...Roots in reality

* One to Many (1) Customer → Orders

⇒ Related data

(2) Student → ^{Date Exam} Paper(s)

* Real world data is Messy & Interrelated

* Relationships and Joins

⇒ Types of Data Relationships

→ (1) One to Many → Avoid duplication

→ Customers & Orders

Foreign key → Reference to another table

foreign key (cust_id) References Customer (id)

* Cross Join ⇒ Implicit join

select * from orders where customerid =


C

select id from Customer

where lastname = 'George'

Imp

Comments
Date: / /

* Inner Join \Rightarrow 

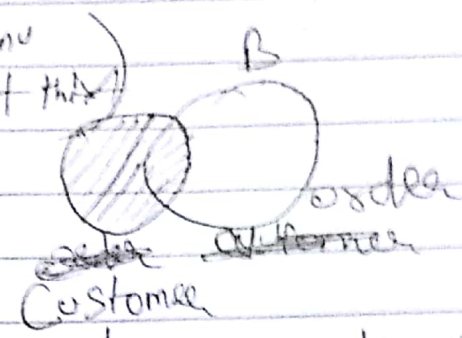
Join orders
Where $customer_id = orders_id$



* Left Join \Rightarrow (only support this)
Dev Env

Left join orders

On $customer_id = orders_customer_id$



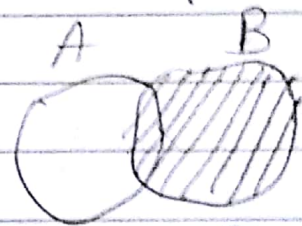
* IF NULL

$IFNULL(Sum(amount), 0)$

Case
When then
else
end as (header
column)

* RIGHT JOIN

Select
 $IFNULL(firstname, 'missing')$ As First.



* On Delete Cascade \Rightarrow Foreign Key constraint

\rightarrow foreign key (student_id)
reference student(id)

*] Many to Many

Books ↔ Authors
Students ↔ Classes.

Date: _____
Inner Join
↳ Overlap

TV Series, Reviewers, Reviews.

(A) Reviewers (B) series (C) Reviews
id
↓ Reviewers_id
Series_id.
Rating

TV Join

→ from series

Join reviews

On series.id = reviews.series_id;

Creating our tables

→ Reviewers

(Inner Join)

Join reviews

On reviewer.id = reviews_reviewer_id;

* Rounding off ⇒ Round (Avg (rating))
2

,) AS Avg_Rating

SQL

Instagram Database Clone

Date: _____

Varchar (255) → User
→ Photos ⇒ Schema
→ Comments
→ Likes
⇒ Followers

→ Hash tags → Tag Photo tag

⇒ Jumbo Dataset

↳ Order by created-at.

↳ Day name (created-at) As day
Count (*) As total
from users

Group by day
order by total Desc;
limit 2;

Left

Inner join photos
on users.id = photos.user_id,

where photos.id is NULL;

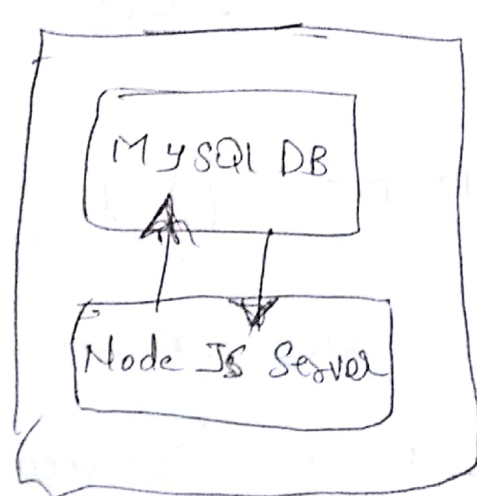
Select
(Select count (*) from users) /
Photos; } date: _____

Introduction to Node → Node JS

{ select 1/2;
05

PHP, Java, Python, Node, Ruby, C#
Javascript.

```
node filename.js  
console.log(".....!");
```

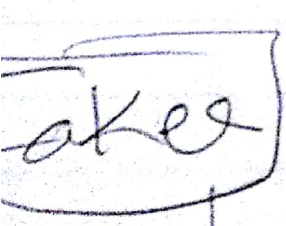


Introducing Faker → Packages

Nodes ⇒ 1000's of libraries

NPM → Node package Manager.

```
faker.internet.email() | faker.date.past
```



↓
Email

generate userdata for us

console.log(faker.internet.email());

↳ To execute

(faker.address.city()),
(" " streetaddress()),

→ function generateAddress()
{

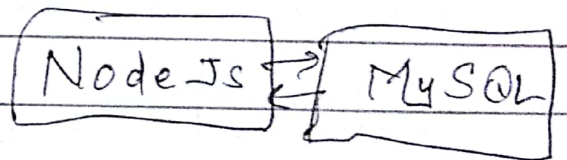
console.log(faker.address.streetaddress());
console.log(faker.address.city());
console.log(faker.address.state());

}

Execute ⇒ generateAddress();

⇒ Generate Data

* MySQL Package



⇒ How many users

npm install mysql

on js (node.js / javascript)

=> var mysql = require('mysql')

var Connection = mysql.createConnection

host : "vishal007"

Date: _____

user :

database : Join-us

;

);

* Users table (Schema.sql)

```
create table users (  
  email varchar(255) primary key,  
  created_at timestamp default now()  
);
```

Selecting using node

insert into users (email)

values ('abc@yahoo.com');

var q = "select * from users"

connection.query(q, function (error, results, fields)

{
 if (error) throw error;

console.log(results);

};

Wings of Wonder... Connection.end()

MySQL / Node Magic

WEB APP

→ nodesframework.com

*] NPM Init & Package.json files

NPM init → ^{packages} 5 files, (packages)
 ^{log files} 109 files
 ↳ Starting a new app

NPM install --save

Req → Request
Res → Response

^{HTML} Adding EJS Templates

→ Embedded Java Script (EJS)

→ Integrating MySQL

Database Triggers

Date: _____

→ SQL statements automatically run when a specific table is changed.

Create trigger trigger_name

trigger_name, trigger_event on table_name for
Each row

Begin

End;

Trigger time	Trigger Event	Table name
Before	Insert	Photos
After	Update	Users
	Delete	