Lesson 1

Create database if not exists Users;

http://owu.linkpc.net/ - databases okten

Show databases  
use назва бази даних  
show tables

create table **users**  
(  
 id int not null primary key auto\_increment, - первинний ключ, унікальний

Name varchar(255) not null, - varchar - стрінга

Age int not null,

Gender varcgar(6) npt nul,

Weight double(5,2), - 5 символі дрібна частина 2 (97.55)

bDay datetime , - “yyyy-m-d” – делімітр, розподілювач

stamp timestamp, – часова мітка коли зявився в таблиці, в нових версіях має бути дефолт Current\_timestamp on update Current\_timestamp

status boolen (bool, tinyint)  
);

insert into users (ячейки, які будуть заповнюватись) – типу конструктор js  
 values (null,‘max’, 16,’male’),  
 (null,‘max’, 16,’male’);

alter table users  
 modify column stamp timestamp default Current\_timestamp on apdate   
 Current\_timestamp;

alter table user  
 add column ’назва колонки’ int;

select \* from users;

select \* from users where name=’oleg’;

select \* from users where name like ‘o%’;

select \* from users where name like **‘\_i\_a%**’;

select \* from users where age!=19;

select \* from users where age>19 and <=30;

select \* from users where age between 19 and 30 or name=’max’;

select \* from users where age in(12,23);

select \* from users where length(name) >4;

select name, age from users where length(name) >4;

select \* from users order by name; -сортування

select \* from users order by name desc; -навпаки

select \* from user limit 2; перші два юзера (пагінація)

select \* from user where gender=’female’ order by id desc limit 2; -дві останні дівчини

select \* from user limit 2 offset 2; -пропусти 2 візьми інші 2

агрегатні функції (5 штук)

select max(age) as max\_age from users;

select min(age) as min\_age from users;

select count(age) as count, gender from users group by gender;

select avg(age) from users;

select sum(age) from users;

drop schema Users; видаляє базу

drop table user;

delete from users where name like ‘karina%’;

update users set age=2 where id=4;

update users set age=5 where id is null;

Lesson 2

Реліаційна (звязкова) база даних

Один до одного і один до багатьох  
select distinct name from user – унікальні значення  
select max(age) from user  
union  
select min(age) from user; обєднання

Один до одного  
create table users  
(  
 id int primary key auto\_increment,

username varchar(255) not null,

Age varchar(255) not null,

Profile\_id int not null unique

Foreign key (Profile\_id) references profile(id) – прив’язати до ідішки таблиці профіль (каскади)

);

create table profile  
(  
 id int primary key auto\_increment,

name varchar(100) not null,

surname varchar(100) not null,

age int not null default 18  
);

select \* from users;  
select \* from profile;  
select \* from users join profile p on.id = users.profile\_id;

select users.id, users.username, profile.\* from users  
 join profile p on.id = users.profile\_id;

один до багатьох

create table cities  
(  
 id int primary key auto\_increment,

city varchar(100) null  
);

create table users  
(  
 id int primary key auto\_increment,

name varchar(100) not null,

Age int not null,

City\_id int null,

Foreign key (city\_id) references cities(id)

)

select \* from users u join cities c on c.id = u.city\_id;

select \* from users u left join cities c on c.id = u.city\_id;

select \* from users u right join cities c on c.id = u.city\_id;

select \* from client c join app a on c.id = u.city\_id;

select \* from department c   
join dep d on d.iddep = c.dep\_id\_dep  
join app a on c.idCli = a.Cl\_id\_cl  
where City = “lviv”;

Lesson 3

Нормалізація баз даниз  
не може бути масивів в ячейках, через кому

Багато до багатьох, проміжна табличка

Не мають повторюватись, ключові значення виносити в іншу таблицю

Create table users

(  
 id int primary key auto\_increment,

name varchar(100) not null,

Age int not null

)

Create table cars

(  
 id int primary key auto\_increment,

brand varchar(100) not null,

year int not null

);

Create table cars\_users

(  
 user\_id int,

Car\_id int,

Primary key (user\_id, Car\_id),

Foreign key (user\_id) references Cars(id),

Foreign key (car\_id) references users(id)

);

Зробити табл

Обєднуєм 3 таблички