Курсовой проект «Базы данных»

Выполнила Лютова Д.А.

Оглавление

[Описание БД 1](#_Toc28451275)

[Описание таблиц 2](#_Toc28451276)

[Наполнение БД данными 5](#_Toc28451277)

[Внешние ключи 8](#_Toc28451278)

[Индексы 10](#_Toc28451279)

[ERDiagram для БД 11](#_Toc28451280)

[Скрипты выборок, представлений, процедур, триггеров 12](#_Toc28451281)

# Описание БД

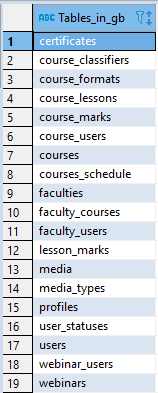
База данных gb.sql описывает модель хранения данных сайта <https://geekbrains.ru/> Это образовательный портал, занимающийся обучением в сфере программирования, веб-дизайна, маркетинга.

База данных содержит таблицы для хранения информации о пользователях, преподавателях, курсах и факультетах сервиса Гикбрейнс. Также база включает данные об оценках и сертификатах пользователей. Для доработки базы может быть добавлена информация о тестах, профессиях, статьях портала, описан форум и раздел вакансий.

Для генерации тестовых данных для таблиц был использован сервис <http://filldb.info>

# Описание таблиц

Для курсового проекта было создано 19 таблиц:



**CREATE** **DATABASE** gb;

**USE** gb;

-- Пользователи

**CREATE** **TABLE** users (

id **INT** **UNSIGNED** **NOT** **NULL** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

first\_name **VARCHAR**(100) **NOT** **NULL**,

last\_name **VARCHAR**(100) **NOT** **NULL**,

created\_at **DATETIME** **DEFAULT** **NOW**(),

updated\_at **DATETIME** **DEFAULT** **NOW**() **ON** **UPDATE** **NOW**()

);

-- Профили

**CREATE** **TABLE** profiles (

user\_id **INT** **UNSIGNED** **NOT** **NULL** **PRIMARY** **KEY**,

status\_id **INT** **UNSIGNED** **NOT** **NULL**,

email **VARCHAR**(120) **NOT** **NULL** **UNIQUE**,

phone **VARCHAR**(120) **NOT** **NULL** **UNIQUE**,

sex **CHAR**(1),

birthday **DATE**,

country **VARCHAR**(100)

hometown **VARCHAR**(100),

photo\_id **INT** **UNSIGNED**,

troika\_kard **TINYINT**,

about **TEXT**,

interests **VARCHAR**(255),

passw **VARCHAR**(20)

);

-- Статусы: студент, наставник, преподаватель

**CREATE** **TABLE** user\_statuses (

id **INT** **UNSIGNED** **NOT** **NULL** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

name **VARCHAR**(100) **NOT** **NULL** **UNIQUE**

);

-- Курсы

**CREATE** **TABLE** courses (

id **INT** **UNSIGNED** **NOT** **NULL** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

heading **VARCHAR**(255) **NOT** **NULL** **UNIQUE**,

course\_format\_id **INT** **UNSIGNED** **NOT** **NULL**,

course\_classifier\_id **INT** **UNSIGNED** **NOT** **NULL**,

duration **VARCHAR**(100) **NOT** **NULL**,

number\_of\_classes **INT** **UNSIGNED** **NOT** **NULL**,

description **TEXT**,

program **TEXT**,

price **DECIMAL**(10,2),

created\_at **DATETIME** **DEFAULT** **NOW**(),

updated\_at **DATETIME** **DEFAULT** **NOW**() **ON** **UPDATE** **NOW**()

);

-- Расписание курсов: ближайший поток, преподаватель

**CREATE** **TABLE** courses\_schedule (

course\_id **INT** **UNSIGNED** **NOT** **NULL**,

nearest\_stream **DATETIME**,

lecturer\_id **INT** **UNSIGNED** **NOT** **NULL**,

**PRIMARY** **KEY** (course\_id, nearest\_stream)

);

-- Форматы курсов: вебинар, интерактивный, видеокурс

**CREATE** **TABLE** course\_formats (

id **INT** **UNSIGNED** **NOT** **NULL** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

name **VARCHAR**(255) **NOT** **NULL** **UNIQUE**,

description **VARCHAR**(255)

);

-- Классификатор курсов: Бесплатные, Для начинающих, Веб-разработка, Разработка программ, Веб-дизайн, Мобильная разработка и т.д.

**CREATE** **TABLE** course\_classifiers (

id **INT** **UNSIGNED** **NOT** **NULL** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

name **VARCHAR**(150) **NOT** **NULL** **UNIQUE**

);

-- Курсы студентов

**CREATE** **TABLE** course\_users (

user\_id **INT** **UNSIGNED** **NOT** **NULL**,

course\_id **INT** **UNSIGNED** **NOT** **NULL**,

is\_finised **BOOLEAN**,

**PRIMARY** **KEY** (course\_id, user\_id)

);

-- Сертификаты

**CREATE** **TABLE** certificates (

id **INT** **UNSIGNED** **NOT** **NULL** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

user\_id **INT** **UNSIGNED** **NOT** **NULL**,

course\_id **INT** **UNSIGNED** **NOT** **NULL**,

created\_at **DATETIME** **DEFAULT** **NOW**()

);

-- Оценка курса и преподавателя

**CREATE** **TABLE** course\_marks (

id **INT** **UNSIGNED** **NOT** **NULL** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

user\_id **INT** **UNSIGNED** **NOT** **NULL**,

course\_id **INT** **UNSIGNED** **NOT** **NULL**

lecturer\_mark **INT** **UNSIGNED**,

course\_mark **INT** **UNSIGNED**,

comment **TEXT**,

created\_at **DATETIME** **DEFAULT** **NOW**()

);

-- Факультеты

**CREATE** **TABLE** faculties (

id **INT** **UNSIGNED** **NOT** **NULL** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

name **VARCHAR**(255) **NOT** **NULL** **UNIQUE**,

description **TEXT**,

program **TEXT**,

price **DECIMAL**(10,2),

created\_at **DATETIME** **DEFAULT** **NOW**(),

updated\_at **DATETIME** **DEFAULT** **NOW**() **ON** **UPDATE** **NOW**()

);

-- Курсы на факультетах

**CREATE** **TABLE** faculty\_courses (

faculty\_id **INT** **UNSIGNED** **NOT** **NULL**,

course\_id **INT** **UNSIGNED** **NOT** **NULL**,

**PRIMARY** **KEY** (course\_id, faculty\_id)

);

-- Студенты факультетов

**CREATE** **TABLE** faculty\_users (

user\_id **INT** **UNSIGNED** **NOT** **NULL**,

faculty\_id **INT** **UNSIGNED** **NOT** **NULL**,

**PRIMARY** **KEY** (user\_id, faculty\_id)

);

-- Содержание уроков

**CREATE** **TABLE** course\_lessons (

id **INT** **UNSIGNED** **NOT** **NULL** **PRIMARY** **KEY**,

course\_id **INT** **UNSIGNED** **NOT** **NULL**,

topic **TINYTEXT**,

video\_id **INT** **UNSIGNED**,

manual\_id **INT** **UNSIGNED**,

presentation\_id **INT** **UNSIGNED**,

home\_tasks **TEXT**

);

-- Оценки студентов за уроки

**CREATE** **TABLE** lesson\_marks (

course\_lesson\_id **INT** **UNSIGNED** **NOT** **NULL** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

user\_id **INT** **UNSIGNED**,

mark **INT** **UNSIGNED**,

created\_at **DATETIME** **DEFAULT** **NOW**()

);

-- Медиафайлы

**CREATE** **TABLE** media (

id **INT** **UNSIGNED** **NOT** **NULL** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

media\_type\_id **INT** **UNSIGNED** **NOT** **NULL**,

user\_id **INT** **UNSIGNED** **NOT** **NULL**,

filename **VARCHAR**(255) **NOT** **NULL**,

**size** **INT** **NOT** **NULL**,

metadata JSON,

created\_at **DATETIME** **DEFAULT** **CURRENT\_TIMESTAMP**,

updated\_at **DATETIME** **DEFAULT** **CURRENT\_TIMESTAMP** **ON** **UPDATE** **CURRENT\_TIMESTAMP**

);

-- Таблица типов медиафайлов

**CREATE** **TABLE** media\_types (

id **INT** **UNSIGNED** **NOT** **NULL** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

name **VARCHAR**(255) **NOT** **NULL** **UNIQUE**

);

-- Вебинары

**CREATE** **TABLE** webinars (

id **INT** **UNSIGNED** **NOT** **NULL** **AUTO\_INCREMENT** **PRIMARY** **KEY**,

heading **VARCHAR**(255) **NOT** **NULL** **UNIQUE**,

classifier\_id **INT** **UNSIGNED** **NOT** **NULL**,

description **TEXT**,

`date` **DATETIME**,

lecturer\_id **INT** **UNSIGNED** **NOT** **NULL**

created\_at **DATETIME** **DEFAULT** **NOW**(),

updated\_at **DATETIME** **DEFAULT** **NOW**() **ON** **UPDATE** **NOW**()

);

-- Слушатели вебинаров

**CREATE** **TABLE** webinar\_users (

webinar\_id **INT** **UNSIGNED** **NOT** **NULL**,

user\_id **INT** **UNSIGNED** **NOT** **NULL**,

**PRIMARY** **KEY** (webinar\_id, user\_id)

);

# Наполнение БД данными

Для генерации тестовых данных для таблиц был использован сервис <http://filldb.info>

Тестовые данные сохранены в файле fulldb27-12-2019 13-16

После генерации данных некоторые поля нуждались в доработке.

Были внесены следующие изменения:

**USE** gb;

-- Устанавливаем время создания раньше времени обновления в таблице users

**UPDATE** users **SET** created\_at = updated\_at, updated\_at = created\_at

**WHERE** created\_at > updated\_at;

-- Корректируем обозначение пола, номера карты тройка, photo\_id в таблице profiles

**CREATE** **TEMPORARY** **TABLE** sex (sex **CHAR**(1));

**INSERT** **INTO** sex **VALUES** ('m'), ('f');

**UPDATE** profiles **SET** sex = (**SELECT** sex **FROM** sex **ORDER** **BY** **RAND**() **LIMIT** 1);

**UPDATE** profiles **SET** troika\_kard = **FLOOR**(1 + (**RAND**() \* 200));

**UPDATE** profiles **SET** photo\_id = **FLOOR**(1 + (**RAND**() \* 200));

-- Обновляем название статусов пользователей

**TRUNCATE** user\_statuses;

**INSERT** **INTO** user\_statuses (name) **VALUES**

('Студент'),

('Наставник'),

('Преподаватель')

;

-- Устанавливаем время создания раньше времени обновления в таблице courses

**UPDATE** courses **SET** created\_at = updated\_at, updated\_at = created\_at

**WHERE** created\_at > updated\_at;

-- Меняем данные: ставим в таблицу courses\_schedul в поле lecturer\_id тех пользователей, чей статус 'Преподаватель'

**UPDATE** courses\_schedule **SET** lecturer\_id =

(**SELECT** users.id **FROM** users

**JOIN** profiles

**ON** users.id = profiles.user\_id

**JOIN** user\_statuses

**ON** profiles.status\_id = user\_statuses.id

**WHERE** name = 'Преподаватель'

**ORDER** **BY** **RAND**()

**LIMIT** 1

);

-- Устанавливаем в поле user\_id тех студентов, у кого метка курса is\_finished = TRUE

**UPDATE** certificates **SET** user\_id =

(**SELECT** users.id **FROM** users

**JOIN** course\_users

**ON** course\_users.user\_id = users.id

**WHERE** course\_users.is\_finished = 1

**ORDER** **BY** **RAND**() **LIMIT** 1

);

-- Устанавливаем в поле user\_id тех студентов, и те курсы, где метка курса is\_finished = TRUE

**UPDATE** certificates **SET** user\_id = (

**SELECT** users.id **FROM** users

**JOIN** course\_users

**ON** course\_users.user\_id = users.id

**WHERE** course\_users.is\_finished = 1

**ORDER** **BY** **RAND**() **LIMIT** 1),

course\_id = (

**SELECT** courses.id **FROM** courses

**JOIN** course\_users

**ON** course\_users.course\_id = courses.id

**WHERE** course\_users.is\_finished = 1

**ORDER** **BY** **RAND**() **LIMIT** 1

);

-- Устанавливаем в поле user\_id тех студентов, и те курсы, где метка курса is\_finished = TRUE

**UPDATE** course\_marks **SET** user\_id = (

**SELECT** users.id **FROM** users

**JOIN** course\_users

**ON** course\_users.user\_id = users.id

**WHERE** course\_users.is\_finished = 1

**ORDER** **BY** **RAND**() **LIMIT** 1),

course\_id = (

**SELECT** courses.id **FROM** courses

**JOIN** course\_users

**ON** course\_users.course\_id = courses.id

**WHERE** course\_users.is\_finished = 1

**ORDER** **BY** **RAND**() **LIMIT** 1

);

-- Оценки курса от 1 до 5

**UPDATE** course\_marks **SET** course\_mark = **FLOOR**(1 + (**RAND**() \* 5))

-- Устанавливаем id медиа-файлов рандомно

**UPDATE** course\_lessons **SET** video\_id = **FLOOR**(1 + (**RAND**() \* 200));

**UPDATE** course\_lessons **SET** manual\_id = **FLOOR**(1 + (**RAND**() \* 200));

**UPDATE** course\_lessons **SET** presentation\_id = **FLOOR**(1 + (**RAND**() \* 200));

-- Оценки курса от 1 до 5

**UPDATE** lesson\_marks **SET** mark = **FLOOR**(1 + (**RAND**() \* 5))

-- Меняем id студентов

**UPDATE** lesson\_marks **SET** user\_id = **FLOOR**(1 + (**RAND**() \* 30))

**TRUNCATE** media\_types;

--

**INSERT** **INTO** media\_types (name) **VALUES**

('photo'),

('video'),

('presentation'),

('manual')

;

**UPDATE** media **SET** media\_type\_id = **FLOOR**(1 + (**RAND**() \* 4));

-- Меняем значения в таблице media

**UPDATE** media **SET** user\_id = **FLOOR**(1 + (**RAND**() \* 100));

**UPDATE** media **SET** metadata = **CONCAT**('{"', filename, '":"', **size**, '"}');

**ALTER** **TABLE** media **MODIFY** **COLUMN** metadata JSON;

-- В таблице вебинаров ставим в поле lecturer\_id тех пользователей, чей статус 'Преподаватель'

**UPDATE** webinars **SET** lecturer\_id =

(**SELECT** users.id **FROM** users

**JOIN** profiles

**ON** users.id = profiles.user\_id

**JOIN** user\_statuses

**ON** profiles.status\_id = user\_statuses.id

**WHERE** name = 'Преподаватель'

**ORDER** **BY** **RAND**()

**LIMIT** 1

);

**UPDATE** webinars **SET** classifier\_id = **FLOOR**(1 + (**RAND**() \* 10));

# Внешние ключи

**USE** gb;

**ALTER** **TABLE** profiles

**ADD** **CONSTRAINT** profiles\_user\_id\_fk

**FOREIGN** **KEY** (user\_id) **REFERENCES** users(id)

**ON** **DELETE** **CASCADE**;

**ALTER** **TABLE** profiles

**ADD** **CONSTRAINT** profiles\_status\_id\_fk

**FOREIGN** **KEY** (status\_id) **REFERENCES** user\_statuses(id);

**ALTER** **TABLE** courses **CHANGE** **COLUMN** course\_format\_id format\_id **INT** **UNSIGNED** **NOT** **NULL**;

**ALTER** **TABLE** courses **CHANGE** **COLUMN** course\_classifier\_id classifier\_id **INT** **UNSIGNED** **NOT** **NULL**;

**ALTER** **TABLE** courses

**ADD** **CONSTRAINT** courses\_format\_id\_fk

**FOREIGN** **KEY** (format\_id) **REFERENCES** course\_formats(id),

**ADD** **CONSTRAINT** courses\_classifier\_id\_fk

**FOREIGN** **KEY** (classifier\_id) **REFERENCES** course\_classifiers(id);

**ALTER** **TABLE** courses\_schedule

**ADD** **CONSTRAINT** courses\_schedule\_course\_id\_fk

**FOREIGN** **KEY** (course\_id) **REFERENCES** courses(id),

**ADD** **CONSTRAINT** courses\_schedule\_lecturer\_id\_fk

**FOREIGN** **KEY** (lecturer\_id) **REFERENCES** users(id);

**ALTER** **TABLE** course\_users

**ADD** **CONSTRAINT** course\_users\_user\_id\_fk

**FOREIGN** **KEY** (user\_id) **REFERENCES** users(id)

**ON** **DELETE** **CASCADE**,

**ADD** **CONSTRAINT** course\_users\_course\_id\_fk

**FOREIGN** **KEY** (course\_id) **REFERENCES** courses(id)

**ON** **DELETE** **CASCADE**;

**ALTER** **TABLE** certificates

**ADD** **CONSTRAINT** certificates\_course\_id\_fk

**FOREIGN** **KEY** (course\_id) **REFERENCES** courses(id),

**ADD** **CONSTRAINT** certificates\_user\_id\_fk

**FOREIGN** **KEY** (user\_id) **REFERENCES** users(id);

**ALTER** **TABLE** course\_marks

**ADD** **CONSTRAINT** course\_marks\_course\_id\_fk

**FOREIGN** **KEY** (course\_id) **REFERENCES** courses(id),

**ADD** **CONSTRAINT** course\_marks\_user\_id\_fk

**FOREIGN** **KEY** (user\_id) **REFERENCES** users(id);

**ALTER** **TABLE** faculty\_courses

**ADD** **CONSTRAINT** faculty\_courses\_faculty\_id\_fk

**FOREIGN** **KEY** (faculty\_id) **REFERENCES** faculties(id)

**ON** **DELETE** **CASCADE**,

**ADD** **CONSTRAINT** faculty\_courses\_course\_id\_fk

**FOREIGN** **KEY** (course\_id) **REFERENCES** courses(id)

**ON** **DELETE** **CASCADE**;

**ALTER** **TABLE** faculty\_users

**ADD** **CONSTRAINT** faculty\_users\_faculty\_id\_fk

**FOREIGN** **KEY** (faculty\_id) **REFERENCES** faculties(id),

**ADD** **CONSTRAINT** faculty\_users\_user\_id\_fk

**FOREIGN** **KEY** (user\_id) **REFERENCES** users(id)

**ON** **DELETE** **CASCADE**;

**ALTER** **TABLE** course\_lessons

**ADD** **CONSTRAINT** course\_lessons\_course\_id\_fk

**FOREIGN** **KEY** (course\_id) **REFERENCES** courses(id)

**ON** **DELETE** **CASCADE**;

**ALTER** **TABLE** lesson\_marks

**ADD** **CONSTRAINT** lesson\_marks\_course\_lesson\_id\_fk

**FOREIGN** **KEY** (course\_lesson\_id) **REFERENCES** course\_lessons(id)

**ON** **DELETE** **CASCADE**,

**ADD** **CONSTRAINT** lesson\_marks\_user\_id\_fk

**FOREIGN** **KEY** (user\_id) **REFERENCES** users(id)

**ON** **DELETE** **CASCADE**;

**ALTER** **TABLE** media

**ADD** **CONSTRAINT** media\_type\_id\_fk

**FOREIGN** **KEY** (media\_type\_id) **REFERENCES** media\_types(id),

**ADD** **CONSTRAINT** media\_user\_id\_fk

**FOREIGN** **KEY** (user\_id) **REFERENCES** users(id)

**ON** **DELETE** **CASCADE**;

**ALTER** **TABLE** webinars

**ADD** **CONSTRAINT** webinars\_classifier\_id\_fk

**FOREIGN** **KEY** (classifier\_id) **REFERENCES** course\_classifiers(id),

**ADD** **CONSTRAINT** webinars\_lecturer\_id\_fk

**FOREIGN** **KEY** (lecturer\_id) **REFERENCES** users(id);

**ALTER** **TABLE** webinar\_users

**ADD** **CONSTRAINT** webinar\_users\_webinar\_id\_fk

**FOREIGN** **KEY** (webinar\_id) **REFERENCES** webinars(id),

**ADD** **CONSTRAINT** webinar\_users\_user\_id\_fk

**FOREIGN** **KEY** (user\_id) **REFERENCES** users(id);

-- Не удается связать таблицы с медиа

**ALTER** **TABLE** course\_lessons

**ADD** **CONSTRAINT** course\_lessons\_video\_id\_fk

**FOREIGN** **KEY** (video\_id) **REFERENCES** media(id)

**ON** **DELETE** **SET** **NULL**,

**ADD** **CONSTRAINT** course\_lessons\_manual\_id\_fk

**FOREIGN** **KEY** (manual\_id) **REFERENCES** media(id)

**ON** **DELETE** **SET** **NULL**,

**ADD** **CONSTRAINT** course\_lessons\_presentation\_id\_fk

**FOREIGN** **KEY** (presentation\_id) **REFERENCES** media(id)

**ON** **DELETE** **SET** **NULL**;

**ALTER** **TABLE** profiles

**ADD** **CONSTRAINT** profiles\_photo\_id\_fk

**FOREIGN** **KEY** (photo\_id) **REFERENCES** media(id)

**ON** **DELETE** **SET** **NULL**;

# Индексы

Для оптимизации запросов были проиндексированы столбцы таблиц пользователи, профили, курсы, факультеты и медиа:

**CREATE** **INDEX** users\_id\_idx **ON** users(id);

**CREATE** **INDEX** profiles\_user\_id\_status\_id\_idx **ON** profiles(user\_id, status\_id);

**CREATE** **INDEX** profiles\_user\_id\_passw\_idx **ON** profiles(user\_id, passw);

**CREATE** **INDEX** courses\_id\_idx **ON** courses(id);

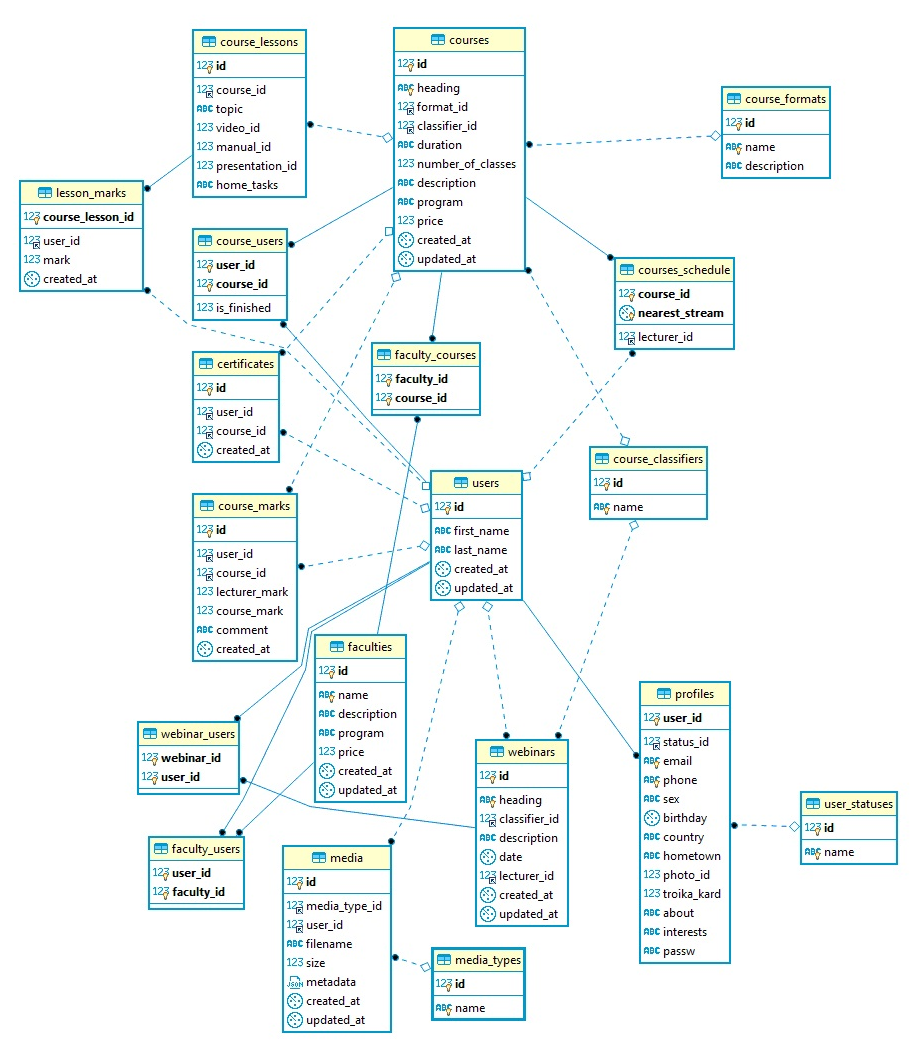
**CREATE** **INDEX** course\_users\_user\_id\_course\_id\_idx **ON** course\_users(user\_id, course\_id);

**CREATE** **INDEX** faculties\_id\_idx **ON** faculties(id);

**CREATE** **INDEX** faculty\_courses\_faculty\_id\_course\_id\_idx **ON** faculty\_courses(faculty\_id, course\_id);

**CREATE** **INDEX** media\_user\_id\_media\_type\_id\_idx **ON** media(user\_id, media\_type\_id);

# ERDiagram для БД



# Скрипты выборок, представлений, процедур, триггеров

**USE** gb;

-- Количество студентов на курсе

**SELECT** courses.heading, **COUNT**(course\_users.user\_id) **AS** amount

**FROM** course\_users

**JOIN** courses

**ON** courses.id = course\_users.course\_id

**GROUP** **BY** courses.heading;

-- Средняя оценка преподавателя

**SELECT** users.id, users.first\_name, users.last\_name, **AVG**(course\_marks.lecturer\_mark) **AS** rating

**FROM** users

**JOIN** course\_marks

**ON** users.id = course\_marks.user\_id

**JOIN** profiles

**ON** users.id = profiles.user\_id

**JOIN** user\_statuses

**ON** profiles.status\_id = user\_statuses.id

**WHERE** name = 'Преподаватель'

**GROUP** **BY** users.id

**ORDER** **BY** rating **DESC**;

-- Количество незавершенных курсов у студента

**SELECT** users.id, users.first\_name, users.last\_name, **COUNT**(course\_users.course\_id) **AS** amount

**FROM** users

**JOIN** course\_users

**ON** users.id = course\_users.user\_id

**JOIN** profiles

**ON** users.id = profiles.user\_id

**JOIN** user\_statuses

**ON** profiles.status\_id = user\_statuses.id

**WHERE** name = ('Студент' **OR** 'Наставник') **AND** course\_users.is\_finished = 0

**GROUP** **BY** users.id;

-- Все преподаватели

**CREATE** **OR** **REPLACE** **VIEW** lecturers

**AS** **SELECT** id, first\_name, last\_name

**FROM** users **WHERE** id **IN**

(**SELECT** user\_id **FROM** profiles

**WHERE** status\_id =

(**SELECT** id **FROM** user\_statuses

**WHERE** name = 'Преподаватель'))

**ORDER** **BY** id;

**SELECT** \* **FROM** lecturers;

-- Разбивка студентов по факультетам

**CREATE** **OR** **REPLACE** **VIEW** students

**AS** **SELECT** users.id, users.first\_name, users.last\_name, faculties.name

**FROM** users

**JOIN** faculty\_users

**ON** users.id = faculty\_users.user\_id

**JOIN** faculties

**ON** faculties.id = faculty\_users.faculty\_id

**ORDER** **BY** faculties.name;

**SELECT** \* **FROM** students;

-- Процедура: поиск пользователя по городу

**DELIMITER -**

**CREATE** **PROCEDURE** get\_user (city\_arg **VARCHAR**(45))

**BEGIN**

**SELECT** user\_id, hometown

**FROM** profiles

**WHERE** hometown = city\_arg;

**END**-

**DELIMITER ;**

**CALL** get\_user('West Jalon');

-- Процедура: id и название вебинара, где количество слушателей больше указанного в качестве аргумента

**DELIMITER -**

**CREATE** **PROCEDURE** info\_webinars (arg **int**)

**BEGIN**

**SELECT** webinar\_users.webinar\_id, webinars.heading, **COUNT**(webinar\_users.user\_id) **AS** amount

**FROM** webinar\_users

**JOIN** webinars

**ON** webinar\_users.webinar\_id = webinars.id

**GROUP** **BY** webinar\_users.webinar\_id

**HAVING** **COUNT**(webinar\_users.user\_id) > arg;

**END**-

**DELIMITER ;**

**CALL** info\_webinars(8);

-- Триггер: добавление новых пользователей

**DELIMITER -**

**CREATE** **TRIGGER** users\_count **AFTER** **INSERT** **ON** users

**FOR** **EACH** **ROW**

**BEGIN**

**SELECT** **COUNT**(\*) **INTO** @total **FROM** users;

**END**-

**DELIMITER ;**

**INSERT** **INTO** users (first\_name, last\_name) **VALUES** ('Дарья', 'Лютова');

**SELECT** @total;

-- Триггер: повышение цены на факультетах на 20%

**DELIMITER -**

**CREATE** **TRIGGER** Before\_Update\_price

**BEFORE** **UPDATE** **ON** faculties

**FOR** **EACH** **ROW**

**BEGIN**

**SET** **NEW**.price = **NEW**.price \* 1.2;

**END**-

**DELIMITER ;**