

Базы данных и SQL (семинары)

Урок 1. Установка СУБД, подключение к БД, просмотр и создание таблиц

Для решения данного домашнего задания вам необходимо воспользоваться сервисом автоматической проверки написанного кода.

Задача 1. Создание сущности с мобильными телефонами.

Создайте таблицу (сущность) с мобильными телефонами `mobile_phones`. При создании необходимо использовать DDL-команды.

Перечень полей (атрибутов):

`id` – числовой тип, автоинкремент, первичный ключ;

`product_name` – строковый тип, обязательный к заполнению;

`manufacturer` – строковый тип, обязательный к заполнению;

`product_count` – числовой тип, беззнаковый;

`price` – числовой тип, беззнаковый.

Используя CRUD-операцию `INSERT`, наполните сущность `mobile_phones` данными:

product_name	manufacturer	product_count	price
iPhone X	Apple	156	76000
iPhone 8	Apple	180	51000
Galaxy S9	Samsung	21	56000
Galaxy S8	Samsung	124	41000
P20 Pro	Huawei	341	36000

-- При написании запросов указывайте не только имя таблицы, но и схему.

-- Название вашей схемы - `itresume7605981`

-- Например, `itresume7605981.tablename`

-- Вы работаете с PostgreSQL

Задача 2. Выборка мобильных телефонов по количеству.

Имеется таблица (сущность) с мобильными телефонами `mobile_phones`.

У сущности имеются следующие поля(атрибуты):

`id` – идентификатор;

`product_name` – название;

`manufacturer` – производитель;

`product_count` – количество;

Необходимо вывести название, производителя и цену для мобильных телефонов, у которых количество больше чем 2.

id product_name manufacturer product_count price -- ----- ----- ----- -
--- 1 iPhone X Apple 156 76000 2 iPhone 8 Apple 180 51000 3 Galaxy S9

Samsung | 21 | 56000 | | 4 | Galaxy S8 | Samsung | 124 | 41000 | | 5 | P20 Pro | Huawei | 341 | 36000
|

Создайте таблицу (сущность) с заказами manufacturer. При создании необходимо использовать DDL-команды.

Перечень полей (атрибутов):

id – числовой тип, автоинкремент, первичный ключ;

name – строковый тип;

Используя CRUD-операцию INSERT, наполните сущность manufacturer в соответствии с данными, имеющимися в атрибуте manufacturer сущности mobile_phones.

-- При написании запросов указывайте не только имя таблицы, но и схему.

-- Название вашей схемы - itresume7605981

-- Например, itresume7605981.tablename

-- Вы работаете с PostgreSQL

Задача 2. Вывод статуса количества мобильных телефонов.

Имеется таблица (сущность) с мобильными телефонами mobile_phones.

У сущности имеются следующие поля(атрибуты):

id – идентификатор;

product_name – название;

manufacturer – производитель;

product_count – количество;

price – цена.

Сущность mobile_phones имеет следующие записи:

id	product_name	manufacturer	product_count	price
1	iPhone X	Apple	156	76000
2	iPhone 8	Apple	180	51000
3	Galaxy S9	Samsung	21	56000
4	Galaxy S8	Samsung	124	41000
5	P20 Pro	Huawei	341	36000

Статусы количества мобильных телефонов (в зависимости от количества): меньше 100 – "little"; от 100 до 300 – "many"; больше 300 – "lots".

Необходимо вывести название, производителя и статус количества для мобильных телефонов.

-- Вы работаете с MySQL

Задача 3. Создание внешнего ключа.

Имеется таблица (сущность) с мобильными телефонами `mobile_phones`.

У сущности имеются следующие поля(атрибуты):

`id` – идентификатор;

`product_name` – название;

`manufacturer` – производитель;

`product_count` – количество;

`price` – цена.

Сущность `mobile_phones` имеет следующие записи:

id	product_name	manufacturer	product_count	price
1	iPhone X	Apple	156	76000
2	iPhone 8	Apple	180	51000
3	Galaxy S9	Samsung	21	56000
4	Galaxy S8	Samsung	124	41000
5	P20 Pro	Huawei	341	36000

Имеется таблица-справочник (сущность) производителей `manufacturer`.

У сущности имеются следующие поля(атрибуты):

`id` – идентификатор;

`name` – название.

Сущность `manufacturer` имеет следующие записи:

id	name
1	Apple
2	Samsung
3	Huawei

Создайте для сущности mobile_phones внешний ключ manufacturer_id (идентификатор производителя), направленный на атрибут id сущности manufacturer. Установите каскадное обновление - CASCADE, а при удалении записи из сущности manufacturer – SET NULL.

Используя CRUD-операцию UPDATE обновите данные в атрибуте manufacturer_id сущности mobile_phones согласно значений, имеющихся в атрибуте manufacturer.

Удалите атрибут manufacturer из сущности mobile_phones.

Выведите идентификатор, название и идентификатор производителя сущности mobile_phones.

-- При написании запросов указывайте не только имя таблицы, но и схему.

-- Название вашей схемы - itresume7605981

-- Например, itresume7605981.tablename

-- Вы работаете с PostgreSQL

Задача 4. Вывод подробного описания статуса.

Имеется таблица (сущность) с заказами orders.

У сущности имеются следующие поля(атрибуты):

id – идентификатор;

mobile_phones_id – идентификатор мобильного телефона;

order_status - статус.

id	mobile_phones_id	order_status
1	1	OPEN
2	1	OPEN
3	1	CLOSED
4	4	OPEN
5	4	CANCELLED

Подробное описание статусов заказа:

OPEN – "Order is in open state" ;

CLOSED - "Order is closed";

CANCELLED - "Order is cancelled"

Необходимо вывести идентификатор и подробное описание статуса заказа.

-- Вы работаете с MySQL

Урок 3. SQL – выборка данных, сортировка, агрегатные функции

Для решения данного домашнего задания вам необходимо воспользоваться сервисом автоматической проверки написанного кода.

Задача 1. Сортировка данных.

Имеется таблица (сущность) с персоналом staff.

У сущности имеются следующие поля(атрибуты):

id – идентификатор;

firstname – имя;

lastname - фамилия.

post - должность,

seniority – стаж;

salary – заработная плата;

age - возраст.

Необходимо вывести идентификатор, имя, фамилию, заработную плату из сущности staff при этом данные должны быть отсортированы в порядке возрастания заработной платы.

-- Вы работаете с MySQL

Задача 2. Сортировка данных.

Имеется таблица (сущность) с персоналом staff.

У сущности имеются следующие поля(атрибуты):

id – идентификатор;

firstname – имя;

lastname - фамилия

post - должность,

seniority – стаж;

salary – заработная плата;

age - возраст.

Необходимо вывести идентификатор, имя, фамилию, заработную плату из сущности staff при этом данные должны быть отсортированы в порядке убывания заработной платы.

-- Вы работаете с MySQL

Задача 3. Максимальная заработная плата.

Имеется таблица (сущность) с персоналом staff.

У сущности имеются следующие поля(атрибуты):

id – идентификатор;

firstname – имя;

lastname - фамилия

post - должность,

seniority – стаж;

salary – заработная плата;

age - возраст.

Необходимо вывести идентификатор, имя, фамилию, заработную плату пяти самых высокооплачиваемых сотрудников из сущности staff.

-- Вы работаете с MySQL

Задача 4. Сортировка данных.

Имеется таблица (сущность) с персоналом staff.

У сущности имеются следующие поля(атрибуты):

id – идентификатор;

firstname – имя;

lastname - фамилия

post - должность,

seniority – стаж;

salary – заработная плата;

age - возраст.

Посчитайте и выведите суммарную зарплату (salary) по каждой специальности (post) из сущности staff.

Порядок вывода атрибутов: должность, суммарная зарплата.

-- Вы работаете с MySQL

Задача 5. Количество сотрудников определенного возраста.

Имеется таблица (сущность) с персоналом staff.

У сущности имеются следующие поля(атрибуты):

id – идентификатор;

firstname – имя;

lastname - фамилия

post - должность,

seniority – стаж;

salary – заработная плата;

age - возраст.

Посчитайте и выведите количество сотрудников с должностью 'Рабочий' и возрастом не моложе 24 лет и не старше 49 лет.

-- Вы работаете с MySQL

Задача 6. Количество специальностей.

Имеется таблица (сущность) с персоналом staff.

У сущности имеются следующие поля(атрибуты):

id – идентификатор;

firstname – имя;

lastname - фамилия

post - должность,

seniority – стаж;

salary – заработная плата;

age - возраст.

Посчитайте и выведите количество уникальных должностей, имеющих у сотрудников в сущности 'staff'.

-- Вы работаете с MySQL

Задача 7. Сотрудники младше 30 лет.

Имеется таблица (сущность) с персоналом staff.

У сущности имеются следующие поля(атрибуты):

id – идентификатор;

firstname – имя;

lastname - фамилия

post - должность,

seniority – стаж;

salary – заработная плата;

age - возраст.

Найдите средний возраст сотрудников по каждой должности из сущности staff.

Выведите только те должности, у которых средний возраст меньше 30 лет.

-- Вы работаете с MySQL

Урок 4. SQL – работа с несколькими таблицами.

Задача 1. Лайки.

Имеется база данных – социальная сеть.

База данных содержит сущности:

users – пользователи;

messages – сообщения;

friend_requests – заявки на дружбу;

communities – сообщества;

users_communities – пользователи сообществ;

media_types – типы медиа;

media – медиа;

likes – лайки;

profiles – профили пользователя.

У сущности "пользователи" имеются следующие поля(атрибуты):

id – идентификатор;

firstname – имя;

lastname - фамилия;

email - адрес электронной почты.

У сущности "профиль пользователя" имеются следующие поля(атрибуты):

user_id – идентификатор;

gender – пол;

birthday - дата рождения;

photo_id - аватарка;

hometown - город.

Атрибут "пол" сущности "профиль пользователя" может принимать следующие значения:

'f' - женский;

'm' - мужской.

У сущности "лайки" имеются следующие поля(атрибуты): id – идентификатор;

user_id – пользователь, который поставил лайк;

media_id - медиа, который лайкнули.

У сущности "медиа" имеются следующие поля(атрибуты):

id – идентификатор;

user_id – пользователь – владелец медиа;

body - содержимое;

filename – ссылка на файл;

created_at - дата создания;

updated_at - дата последнего обновления.

Найдите общее количество лайков, которые получили пользователи женского пола.

-- Вы работаете с MySQL

Задача 2. Лайки 2.

Имеется база данных "социальная сеть".

База данных содержит таблицы:

users – пользователи;

messages – сообщения;

friend_requests – заявки на дружбу;

communities – сообщества;

users_communities – пользователи сообществ;

media_types – типы медиа;

media – медиа;

likes – лайки;

profiles – профили пользователя.

У таблицы "users" имеются следующие поля(атрибуты):

id – идентификатор;

firstname – имя;

lastname – фамилия;

email – адрес электронной почты.

У таблицы "profiles" имеются следующие поля(атрибуты):

user_id – идентификатор;

gender – пол;

birthday – дата рождения;

photo_id – аватарка;

hometown – город.

Атрибут "gender" таблицы "profiles" может принимать следующие значения:

'f' – женский;

'm' – мужской.

У таблицы "likes" имеются следующие поля(атрибуты):

id – идентификатор;

user_id – пользователь, который поставил лайк;

media_id - медиа, который лайкнули.

Найдите количество лайков, которые поставили пользователи женского пола и мужского пола.

Выведите название пола (преобразовав значение атрибута "f" в женский, а значение "m" - мужской) и количество, поставленных лайков, применив к количеству лайков сортировку по убыванию.

-- Вы работаете с MySQL

Задача 3. Кто не отправлял сообщения.

Имеется база данных – социальная сеть.

База данных содержит сущности:

users – пользователи;

messages – сообщения;

friend_requests – заявки на дружбу;

communities – сообщества;

users_communities – пользователи сообществ;

media_types – типы медиа;

media – медиа;

likes – лайки;

profiles – профили пользователя.

У сущности «пользователи» имеются следующие поля(атрибуты):

id – идентификатор;

firstname – имя;

lastname - фамилия;

email - адрес электронной почты.

У сущности «сообщения» имеются следующие поля(атрибуты):

id – идентификатор;

from_user_id – отправитель;

to_user_id – получатель;

body - содержимое;

created_at - дата отправки.

Выведите идентификаторы пользователей, которые не отправляли ни одного сообщения.

-- Вы работаете с MySQL

Задача 4. Друзья.

Имеется база данных – социальная сеть.

База данных содержит сущности:

users – пользователи;

messages – сообщения;

friend_requests – заявки на дружбу;

communities – сообщества;

users_communities – пользователи сообществ;

media_types – типы медиа;

media – медиа;

likes – лайки;

profiles – профили пользователя.

У сущности «заявки на дружбу» имеются следующие поля(атрибуты):

initiator_user_id – инициатор;

target_user_id – получатель;

status - статус;

requested_at - дата создания;

updated_at - дата последнего обновления.

У сущности «пользователи» имеются следующие поля(атрибуты):

id – идентификатор;

firstname – имя;

lastname - фамилия;

email - адрес электронной почты.

Друзья — это пользователи у которых имеется соответствующая запись (заявка) в сущности «заявки на дружбу» и в атрибуте status данной сущности указано значение 'approved'.

Найдите количество друзей у каждого пользователя. Выведите для каждого пользователя идентификатор пользователя, фамилию, имя и количество друзей. Сортировка выводимых записей в порядке возрастания идентификатора пользователя.

-- Вы работаете с MySQL

Урок 5. SQL – оконные функции.

Задача 1. Ранжированный список пользователей.

Имеется база данных – социальная сеть.

База данных содержит сущности:

users – пользователи;

messages – сообщения;

friend_requests – заявки на дружбу;

communities – сообщества;

users_communities – пользователи сообществ;

media_types – типы медиа;

media – медиа;

likes – лайки;

profiles – профили пользователя.

У сущности «пользователи» имеются следующие поля(атрибуты):

id – идентификатор;

firstname – имя;

lastname - фамилия;

email - адрес электронной почты.

У сущности «сообщения» имеются следующие поля(атрибуты):

id – идентификатор;

from_user_id – отправитель;

to_user_id – получатель;

body - содержимое;

created_at - дата отправки.

Найти количество сообщений, отправленных каждым пользователей.

В зависимости от количества отправленных сообщений рассчитать ранг пользователей, первое место присвоив пользователю(ям) с наибольшим количеством отправленных сообщений.

Вывести полученный ранг, имя, фамилия, пользователя и кол-во отправленных сообщений. Выводимый список необходимо отсортировать в порядке возрастания ранга.

-- Вы работаете с MySQL

Задача 2. Нахождение разности соседних дат.

Имеется база данных – социальная сеть.

База данных содержит сущности:

users – пользователи;

messages – сообщения;

friend_requests – заявки на дружбу;

communities – сообщества;

users_communities – пользователи сообществ;

media_types – типы медиа;

media – медиа;

likes – лайки;

profiles – профили пользователя.

У сущности «сообщения» имеются следующие поля(атрибуты):

id – идентификатор;

from_user_id – отправитель;

to_user_id – получатель;

body - содержимое;

created_at - дата отправки.

Получите список сообщений, отсортированных по возрастанию даты отправки.

Вычислите разность между соседними значениями дат отправки. Разности выразите в минутах.

Выведите идентификатор сообщения, дату отправки, дату отправки следующего сообщения и разницу даты отправки соседних сообщений.

-- Вы работаете с MySQL

Урок 6. SQL – Транзакции. Временные таблицы, управляющие конструкции, циклы.

Задача 1.

Создайте функцию, которая принимает кол-во сек и формат их в кол-во дней часов. Пример:
123456 -> '1 days 10 hours 17 minutes 36 seconds '

Задача 2.

Выведите только четные числа от 1 до 10. Пример: 2,4,6,8,10.