SQL\_DDL  
**Первая часть.**

Таблица **employees**

1. Создать таблицу employees  
   - id. serial, primary key,  
   - employee\_name. Varchar(50), not null
2. Наполнить таблицу employee 70 строками.

**1) create** **table** employees(

id serial **primary** **key**,

employee\_name **varchar**(50) **unique** **not** **null**);

**2.1) DO** **$$ 2.2) INSERT INTO employees (employee\_name)**

**BEGIN select ‘Сотрудник’ ||generate\_series(1,70);**

**FOR** i **IN** 1..70 **LOOP**

**INSERT** **INTO** employees (employee\_name)

**VALUES** ('Сотрудник ' || i);

**END** **LOOP**;

**END** **$$**;

Таблица **salary**

1. Создать таблицу salary  
   - id. Serial primary key,  
   - monthly\_salary. Int, not null
2. Наполнить таблицу salary 15 строками:  
   - 1000  
   - 1100  
   - 1200  
   - 1300  
   - 1400  
   - 1500  
   - 1600  
   - 1700  
   - 1800  
   - 1900  
   - 2000  
   - 2100  
   - 2200  
   - 2300  
   - 2400  
   - 2500

**3) create** **table** salary(

id serial **primary** **key**,

monthly\_salary int **not** **null**);

**4.1) DO** **$$ 4.2) INSERT INTO salary (monthly\_salary)**

**BEGIN select (1000 + (GENERATE\_SERIES(0,14)\* 100));**

**FOR** i **IN** 1..15 **LOOP**

**INSERT** **INTO** salary (monthly\_salary)

**VALUES** (1000+(i-1)\*100);

**END** **LOOP**;

**END** **$$**;

Таблица **employee\_salary**

1. Создать таблицу employee\_salary **create** **table** employee\_salary(  
   - id. Serial primary key, id serial **primary** **key**,  
   - employee\_id. Int, not null, unique employee\_id int **unique** **not** **null**,

- salary\_id. Int, not null salary\_id int **not** **null**);

1. Наполнить таблицу employee\_salary 40 строками:  
   - в 10 строк из 40 вставить несуществующие employee\_id

| id | employee\_id | salary\_id |
| --- | --- | --- |
| 1 | 3 | 7 |
| 2 | 1 | 4 |
| 3 | 5 | 9 |
| 4 | 40 | 13 |
| 5 | 23 | 4 |
| 6 | 11 | 2 |
| 7 | 52 | 10 |
| 8 | 15 | 13 |
| 9 | 26 | 4 |
| 10 | 16 | 1 |
| 11 | 33 | 7 |
| ... | ... | ... |

INSERT INTO employee\_salary (employee\_id, salary\_id)

SELECT

CASE

WHEN n <= 30 THEN n

ELSE n + 69

END,

n

FROM generate\_series(1, 40) AS n;

Таблица **roles**

1. Создать таблицу roles **create** **table** roles(  
   - id. Serial primary key, id serial **primary** **key**,  
   - role\_name. int, not null, unique role\_name int **unique** **not** **null**,

Поменять тип столба role\_name с ALTER TABLE roles

int на varchar(30) ALTER COLUMN role\_name TYPE varchar(30);

1. Наполнить таблицу roles 20 строками: **INSERT INTO roles (role\_name)**

Values (‘Junior Python developer’),

(‘...’);

| id | role\_name |
| --- | --- |
| 1 | Junior Python developer |
| 2 | Middle Python developer |
| 3 | Senior Python developer |
| 4 | Junior Java developer |
| 5 | Middle Java developer |
| 6 | Senior Java developer |
| 7 | Junior JavaScript developer |
| 8 | Middle JavaScript developer |
| 9 | Senior JavaScript developer |
| 10 | Junior Manual QA engineer |
| 11 | Middle Manual QA engineer |
| 12 | Senior Manual QA engineer |
| 13 | Project Manager |
| 14 | Designer |
| 15 | HR |
| 16 | CEO |
| 17 | Sales manager |
| 18 | Junior Automation QA engineer |
| 19 | Middle Automation QA engineer |
| 20 | Senior Automation QA engineer |

Таблица **roles\_employee**

1. Создать таблицу **roles\_employee create table roles\_employee(**

- id. Serial primary key, **id serial primary key,**

- employee\_id. Int, not null, unique (внешний ключ для таблицы employees, поле id)

employee\_id int REFERENCES employees(id),

- role\_id. Int, not null (внешний ключ для таблицы roles, поле id)

role\_id int REFERENCES roles(id));

1. Наполнить таблицу **roles\_employee** 40 строками:

INSERT INTO roles\_employee (employee\_id, role\_id)

SELECT

(random() \* 70 + 1)::integer,

(random() \* 20 + 1)::integer

FROM generate\_series(1, 40);

| id | employee\_id | role\_id |
| --- | --- | --- |
| 1 | 7 | 2 |
| 2 | 20 | 4 |
| 3 | 3 | 9 |
| 4 | 5 | 13 |
| 5 | 23 | 4 |
| 6 | 11 | 2 |
| 7 | 10 | 9 |
| 8 | 22 | 13 |
| 9 | 21 | 3 |
| 10 | 34 | 4 |
| 11 | 6 | 7 |
| ... | ... | ... |

SQL\_HW\_1

Создать .sql файл в котором под каждой командой напишите sql запрос который выполнит команду.

.sql файл выгружайте на гит и скидывайте ссылки на проверку.

1. Вывести все поля и все строки.

Select \* from tableName;

2. Вывести всех студентов в таблице

Select \* from students;

3. Вывести только Id пользователей

Select id from students;

4. Вывести только имя пользователей

Select name from students;

5. Вывести только email пользователей

Select email from students;

6. Вывести имя и email пользователей

Select name, email from students;

7. Вывести id, имя, email и дату создания пользователей

Select id, name, email, created\_on from students;

8. Вывести пользователей где password 12333

Select \* from students where password = ‘12333’;

9. Вывести пользователей которые были созданы 2021-03-26 00:00:00

Select \* from students where created\_on = ‘2021-03-26 00:00:00’;

10. Вывести пользователей где в имени есть слово Анна

Select \* from students where name like '%Anna%';

11. Вывести пользователей где в имени в конце есть 8

Select \* from students where name like '%8';

12. Вывести пользователей где в имени в есть буква а

Select \* from students where name like '%a%';

13. Вывести пользователей которые были созданы 2021-07-12 00:00:00

Select \* from students where created\_on = ‘2021-07-12 00:00:00’;

14. Вывести пользователей которые были созданы 2021-07-12 00:00:00 и имеют пароль 1m313

**Select** \* **from** students **where** created\_on = '2021-07-12 00:00:00' **and** **password** = '1m313';

15. Вывести пользователей которые были созданы 2021-07-12 00:00:00 и у которых в имени есть слово Andrey

**Select** \* **from** students **where** created\_on = '2021-07-12 00:00:00' **and** **name** like '%Andrey%';

16. Вывести пользователей которые были созданы 2021-07-12 00:00:00 и у которых в имени есть цифра 8

**Select** \* **from** students **where** created\_on = '2021-07-12 00:00:00' **and** **name** like '%8%';

17. Вывести пользователя у которых id равен 110

**Select** \* **from** students **where** id = '110';

18. Вывести пользователя у которых id равен 153

**Select** \* **from** students **where** id = '153';

19. Вывести пользователя у которых id больше 140

**Select** \* **from** students **where** id > '140';

20. Вывести пользователя у которых id меньше 130

**Select** \* **from** students **where** id < '130';

21. Вывести пользователя у которых id меньше 127 или больше 188

**Select** \* **from** students **where** id < '127' or id > '188';

22. Вывести пользователя у которых id меньше либо равно 137

**Select** \* **from** students **where** id <= '137';

23. Вывести пользователя у которых id больше либо равно 137

**Select** \* **from** students **where** id >= '137';

24. Вывести пользователя у которых id больше 180 но меньше 190

**Select** \* **from** students **where** id > '180' and id < '190';

25. Вывести пользователя у которых id между 180 и 190

**Select** \* **from** students **where** id between 180 and 190;

26. Вывести пользователей где password равен 12333, 1m313, 123313

**Select** \* **from** students **where** password in ('12333','1m313','123313');

27. Вывести пользователей где created\_on равен 2020-10-03 00:00:00, 2021-05-19 00:00:00, 2021-03-26 00:00:00

**Select** \* **from** students **where** created\_on in ('2020-10-03 00:00:00','2021-05-19 00:00:00','2021-03-26 00:00:00');

28. Вывести минимальный id

**Select** min(id) **from** students;

29. Вывести максимальный.

**Select** max(id) **from** students;

30. Вывести количество пользователей

**Select** count(\*) **from** students;

31. Вывести id пользователя, имя, дату создания пользователя. Отсортировать по порядку возрастания даты добавления пользоватлеля.

**Select** id, **name**, created\_on **from** students **order** **by** created\_on ;

32. Вывести id пользователя, имя, дату создания пользователя. Отсортировать по порядку убывания даты добавления пользоватлеля.

**Select** id, **name**, created\_on **from** students **order** **by** created\_on **desc** ;

SQL HomeWork 2. Joins

Подключится к

Host: 159.69.151.133

Port: 5056

DB: подключение к той таблице где делали DDL операции

User: подключение к тем пользователем каким делали DDL операции

Pass: 123

Если для какого-то кейса надо сделать дополнительную таблицу, наполнить её данными, то делайте )

1. Вывести всех работников чьи зарплаты есть в базе, вместе с зарплатами.

**Select** e.employee\_name **from** employees e

**join** salary s **on** e.id = s.id

**where** s.monthly\_salary **is** **not** **null**;

2. Вывести всех работников у которых ЗП меньше 2000.

**Select** e.employee\_name **from** employees e

**join** salary s **on** e.id = s.id

**where** s.monthly\_salary **< 2000**;

3. Вывести все зарплатные позиции, но работник по ним не назначен. (ЗП есть, но не понятно кто её получает.)

**select** s.monthly\_salary

**from** salary s

**left** **join** employees e **on** s.id = e.id

**where** e.id **is** **null**;

4. Вывести все зарплатные позиции меньше 2000 но работник по ним не назначен. (ЗП есть, но не понятно кто её получает.)

**Select** e.employee\_name **from** employees e

**join** salary s **on** e.id = s.id

**where** s.monthly\_salary **< 2000**;

5. Найти всех работников кому не начислена ЗП.

**Select** e.employee\_name **from** employees e

**join** salary s **on** e.id = s.id

**where** s.monthly\_salary = 0 ;

6. Вывести всех работников с названиями их должности.

**Select** e.employee\_name, r.role\_name **from** employees e

**join** roles r **on** e.id = r.id;

7. Вывести имена и должность только Java разработчиков.

**Select** e.employee\_name, r.role\_name **from** employees e

**join** roles r **on** e.id = r.id

**where** role\_name **like** '%Java';

8. Вывести имена и должность только Python разработчиков.

**Select** e.employee\_name, r.role\_name **from** employees e

**join** roles r **on** e.id = r.id

**where** role\_name **like** '%Python%';

9. Вывести имена и должность всех QA инженеров.

**Select** e.employee\_name, r.role\_name **from** employees e

**join** roles r **on** e.id = r.id

**where** role\_name **like** '%QA%';

10. Вывести имена и должность ручных QA инженеров.

**Select** e.employee\_name, r.role\_name **from** employees e

**join** roles r **on** e.id = r.id

**where** role\_name **like** '%Manual QA%';

11. Вывести имена и должность автоматизаторов QA

**Select** e.employee\_name, r.role\_name **from** employees e

**join** roles r **on** e.id = r.id

**where** role\_name **like** '%Automation QA%';

12. Вывести имена и зарплаты Junior специалистов

**Select** e.employee\_name, s.monthly\_salary **from** employees e

**join** salary s **on** e.id = s.id

**join** roles r **on** s.id = r.id

**where** role\_name **like** '%Junior%';

13. Вывести имена и зарплаты Middle специалистов

**Select** e.employee\_name, s.monthly\_salary **from** employees e

**join** salary s **on** e.id = s.id

**join** roles r **on** s.id = r.id

**where** role\_name **like** '%Middle%';

14. Вывести имена и зарплаты Senior специалистов

**Select** e.employee\_name, s.monthly\_salary **from** employees e

**join** salary s **on** e.id = s.id

**join** roles r **on** s.id = r.id

**where** role\_name **like** '%Middle%';

15. Вывести зарплаты Java разработчиков

**Select s.monthly\_salary from salary s**

**join roles r on s.id = r.id**

**where role\_name like '%Java%';**

16. Вывести зарплаты Python разработчиков

**Select s.monthly\_salary from salary s**

**join roles r on s.id = r.id**

**where role\_name like '%Python%';**

17. Вывести имена и зарплаты Junior Python разработчиков

**Select** e.employee\_name, s.monthly\_salary **from** employees e

**join** salary s **on** e.id = s.id

**join** roles r **on** s.id = r.id

**where** role\_name **like** '%Junior Python%';

18. Вывести имена и зарплаты Middle JS разработчиков

**Select** e.employee\_name, s.monthly\_salary **from** employees e

**join** salary s **on** e.id = s.id

**join** roles r **on** s.id = r.id

**where** role\_name **like** '%Middle JavaScript%';

19. Вывести имена и зарплаты Senior Java разработчиков

**Select** e.employee\_name, s.monthly\_salary **from** employees e

**join** salary s **on** e.id = s.id

**join** roles r **on** s.id = r.id

**where** role\_name **like** '%Senior Java \_%';

20. Вывести зарплаты Junior QA инженеров

**Select s.monthly\_salary from salary s**

**join roles r on s.id = r.id**

**where role\_name like '%Junior% %QA%';**

21. Вывести среднюю зарплату всех Junior специалистов

**Select avg(s.monthly\_salary) from salary s**

**join roles r on s.id = r.id**

**where role\_name like '%Junior%';**

22. Вывести сумму зарплат JS разработчиков

**Select sum(s.monthly\_salary) from salary s**

**join roles r on s.id = r.id**

**where role\_name like '%JavaScript%';**

23. Вывести минимальную ЗП QA инженеров

**Select min(s.monthly\_salary) from salary s**

**join roles r on s.id = r.id**

**where role\_name like '%QA%';**

24. Вывести максимальную ЗП QA инженеров

**Select max(s.monthly\_salary) from salary s**

**join roles r on s.id = r.id**

**where role\_name like '%QA%';**

25. Вывести количество QA инженеров

**Select** **count**(id) **from** roles

**where** role\_name **like** '%QA%';

26. Вывести количество Middle специалистов.

**Select** **count**(id) **from** roles

**where** role\_name **like** '%Middle%';

27. Вывести количество разработчиков

**Select** **count**(id) **from** roles

**where** role\_name **like** '%developer%';

28. Вывести фонд (сумму) зарплаты разработчиков.

**Select sum(s.monthly\_salary) from salary s**

**join roles r on s.id = r.id**

**where role\_name like '%**developer**%';**

29. Вывести имена, должности и ЗП всех специалистов по возрастанию

**Select** e.employee\_name, r.role\_name, s.monthly\_salary **from** employees e

**join** salary s **on** e.id = s.id

**join** roles r **on** s.id = r.id

**Order by monthly\_salary**;

30. Вывести имена, должности и ЗП всех специалистов по возрастанию у специалистов у которых ЗП от 1700 до 2300

**Select** e.employee\_name, r.role\_name, s.monthly\_salary **from** employees e

**join** salary s **on** e.id = s.id

**join** roles r **on** s.id = r.id

**where** monthly\_salary **between** 1700 **and** 2300

**Order** **by** monthly\_salary;

31. Вывести имена, должности и ЗП всех специалистов по возрастанию у специалистов у которых ЗП меньше 2300

**Select** e.employee\_name, r.role\_name, s.monthly\_salary **from** employees e

**join** salary s **on** e.id = s.id

**join** roles r **on** s.id = r.id

**where** monthly\_salary **<=** 2300

**Order** **by** monthly\_salary;

32. Вывести имена, должности и ЗП всех специалистов по возрастанию у специалистов у которых ЗП равна 1100, 1500, 2000

**Select** e.employee\_name, r.role\_name, s.monthly\_salary **from** employees e

**join** salary s **on** e.id = s.id

**join** roles r **on** s.id = r.id

**where** monthly\_salary **in (1100,1500,2000)**

**Order** **by** monthly\_salary;