**Экзамен по ТБД. PostgreSQL**

**Блок 1. Тесты.**

**За каждый правильный ответ – 1 балл.**

**Вопрос 1**

Какое выражение используется для получения уникальных значений?

Варианты ответов:

SELECT UNIQUE

SELECT DISTINCT+

SELECT DIFFERENT

SELECT MAX

**Вопрос 2**

Сколько записей возвратит следующий запрос:

SELECT 17 UNION SELECT 21 UNION SELECT 17 UNION ALL SELECT 17;

Варианты ответов:

1

2

3+

NULL

Ничего из перечисленного

**Вопрос 3**

Какой результат возвратит код для таблицы:

|  |  |
| --- | --- |
| **id** | **city** |
| 1 | Oslo |
| 2 | New york |
| 3 | *NULL* |
| 4 | Berlin |
| 5 | 5 |

SELECT COUNT (city)

FROM table;

Варианты ответов:

2

3

4+

5

Ничего из перечисленного

**Вопрос 4**

В таблице есть колонка price со следующими значениями: {20, 20, NULL, 50}. Каким будет результат выполнения функции AVG (price)?

Варианты ответов:

20

30+

40

50

Ничего из перечисленного

**Вопрос 5.**

Имеется таблица table (с единственным столбцом value (integer)),

состоящая из следующих 5 строк:

|  |
| --- |
| **value** |
| 5 |
| 5 |
| *NULL* |
| 5 |
| 5 |

Какой результат вернет запрос:

SELECT (avg(value)\*count(\*)) – sum(value) FROM table

Варианты ответов:

-4

0

NULL+

5

Вызовет ошибку, т.к. не указан GROUP BY

Ничего из перечисленного

**Блок 2. Открытые вопросы.**

**За каждый правильный ответ – 2 балла.**

**Вопрос 6.**

Напишите запрос, который из таблицы **customers** выберет только те **email**, у которых есть дубликаты.

Также укажите количество повторений для каждого такого **email**.

SELECT email, COUNT(\*) AS countFROM customersGROUP BY emailHAVING COUNT(\*) > 1;

**Вопрос 7.**

Из каких подмножеств состоит SQL, расшифруйте их?

DDL - Data Definition Language - CREATE, ALTER, DROP

DML - Data Manipulation Language - INSERT, SELECT, UP

DATE, DELETE

DCL - Data Control Language - GRANT, REVOKE, DENY

TCL - Transaction Control Language - COMMIT, ROLLBACK, SAVEPOINT

**Вопрос 8.**

Таблица enrollments содержит некорректные данные с ids между 20 и 100 (включительно).

TABLE enrollments

id INTEGER NOT NULL PRIMARY KEY

year INTEGER NOT NULL

studentId INTEGER NOT NULL

Напишите запрос, который запишет в поле year каждой некорректной записи значение 2015.

UPDATE enrollmentsSET year = 2015WHERE id BETWEEN 20 AND 100

**Вопрос 9.**

Напишите запрос, чтобы получить **id** покупки с максимальной **value** для каждого **customer**.

Например, есть таблица:

id customer value

1 Alex 5

2 Sam 1

3 Sam 3

В результате нужно получить следующее:

id customer value

1 Alex 5

3 Sam 3

SELECT id, customer, value

FROM exam e

WHERE value = (

SELECT MAX(value)

FROM exam

WHERE customer = e.customer

);

**Вопрос 10.**

Что такое Ограничения (Constraints) в SQL и какие виды бывают?

это ограничение которое используется для данных в таблице

Виды:

UNIQUE

PRIMARY KEY

DEFAULT

FOREIGN KEY

NULL

**Бонусный вопрос. Дополнительные баллы – 2.**

Переписать PostgreSQL запрос так, чтобы не использовать оператор **AND**.

Таблица **users**

|  |  |
| --- | --- |
| **id** | **surname** |
| 1 | ‘Aidyn’ |
| 2 | ‘Arman’ |
| 3 | ‘Darhan’ |

PostgreSQL запрос:

*SELECT*

*id,*

*surname*

*FROM users*

*WHERE id = 2 AND surname = ‘Arman’*

Можно обьединить запросы с помощью UNION

SELECT id, surnameFROM usersWHERE id = 2UNIONSELECT id, surnameFROM usersWHERE surname = 'Arman';

**Блок 3. Задача.**

**За полное решение – 9 баллов.**

Имеются две таблицы *workers* и *project\_bonus*. В первой таблице у нас хранится информация о сотрудниках, их позиций и зарплаты. Во второй таблице хранится информация о бонусах за конкретные проекты и id сотрудников, получивших эти бонусы.

Таблица - *workers*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **name** | **role** | **salary** | **birthday** |
| 1 | Айдын | Директор | 100000 | 1985-09-21 |
| 2 | Раушан | Бухгалтер | 75000 | 1994-02-16 |
| 3 | Бахтияр | Аналитик | 80000 | 1990-11-23 |
| 4 | Дмитрий | Программист | 120000 | 1988-02-13 |
| 5 | Жандос | Аналитик | 80000 | 1996-09-13 |
| 6 | Олег | Кладовщик | 25000 | 1992-03-12 |
| 7 | Дастан | Программист | 120000 | 1994-06-04 |

Таблица – *project\_bonus*

|  |  |  |
| --- | --- | --- |
| **id** | **project** | **bonus** |
| 2 | ИнвестКонсалт | 10000 |
| 3 | НормаТв | 2000 |
| 1 | ИнвестКонсалт | 1500 |
| 1 | НормаТв | 24000 |
| 5 | НормаТв | 3000 |
| 7 | НормаТв | 1800 |
| 2 | КазПром | 2000 |
| 10 | КазПром | 7000 |
| 11 | КазПром | 2000 |

Задача 1. Написать запрос, который получит суммы бонусов за проект в разрезе каждого действующего сотрудника. Этот запрос «упаковать» во VIEW и через VIEW показать результат ниже:

CREATE VIEW sum\_bonus AS

SELECT w.name, w.role, SUM(p.bonus) AS sum

FROM workers w

LEFT JOIN project\_bonus p ON w.id = p.id

GROUP BY w.id, w.name, w.role

SELECT name, role, sum

FROM sum\_bonus

WHERE sum IS NOT NULL

ORDER BY sum ASC

Задача 2. Написать запрос, который посчитает среднюю зарплату в разрезе позиций (ролей). Обратите внимание, что результат средней зарплаты должен отобразиться в целочисленном формате (integer).

SELECT role, CAST(ROUND(AVG(salary)) AS INTEGER) AS avg

FROM workersGROUP BY role

