Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра экономической информатики

Лабораторная работа №3

ЯЗЫК SQL. СОЗДАНИЕ ТАБЛИЦ И ОГРАНИЧЕНИЙ НА SQL

Выполнил: студентка гр.

914301 Хеббекулиева Э.

Проверил: Коваленко Д.И.

Минск 2022

### Цель работы

Изучение структурированного языка запросов Transact - SQL, являющегося основой системы программирования SQL Server, и приобретение навыков применение инструментальных средств разработки и программирования объектов создаваемых баз данных. Изучить SQL-операторы для работы с таблицами и индексами. Изучить sql- команды для создания, изменения и удаления таблиц. Изучить используемые в SQL Server типы ограничений. Изучить SQL-операторы для работы с ограничениями.

SELECT COUNT(\*) as count FROM product WHERE storage\_conditions='cold'

#одинаковые записи

SELECT COUNT(DISTINCT type\_payment) as count FROM supplier

#сколько разного в столбце

SELECT SUM(summ) as sum FROM product\_sale

SELECT SUM(summ) as sum FROM product\_sale WHERE id<3

SELECT \*, UPPER(department\_name) as name FROM department

SELECT \*, UPPER(name\_client) as name FROM client

SELECT \*, LOWER(department\_name) as name FROM department

SELECT \*, LOWER(name\_client) as name FROM client

SELECT \*, MONTH(time) as month FROM product\_sale

SELECT \*, YEAR(time) as year FROM product\_sale

SELECT date, SUM(summ) as sum FROM product\_sale GROUP BY date

SELECT date, SUM(amount) as sum FROM product\_sale GROUP BY date

SELECT department\_name, COUNT(amount\_counters) FROM department GROUP BY department\_name HAVING COUNT(amount\_counters) >= 2

SELECT department\_name, COUNT(amount\_seller) FROM department GROUP BY department\_name HAVING COUNT(amount\_seller) > 5

SELECT COUNT(\*) from department HAVING COUNT(\*) > 3;

SELECT COUNT(\*) from client HAVING COUNT(\*) > 3;

SELECT \* FROM department WHERE id>0 ORDER BY amount\_counters

SELECT \* FROM department WHERE id>0 ORDER BY amount\_seller

INSERT INTO client(name\_client, adress, payment) VALUES ('Igor', 'Minsk', 'card')

INSERT INTO client(name\_client, adress, payment) VALUES ('Vasya', 'Minsk', 'card')

UPDATE client SET adress='Barcelona' WHERE id=1

UPDATE client SET adress='Asgabat' WHERE id=4

DELETE FROM client WHERE id=2

DELETE FROM client WHERE id=3

SELECT \* FROM client WHERE name\_client = 'Enesh'

SELECT \* FROM client WHERE name\_client != 'Enesh'

SELECT \* FROM client WHERE id>=2 AND id<=4

SELECT \* FROM client WHERE id=3 OR adress='Paris'

SELECT \* FROM client WHERE id>=2 AND id<=4 OR adress='Paris'

SELECT \* FROM client WHERE id=1 AND id=3 OR adress='Paris'

SELECT \* FROM product, department

SELECT \* FROM product, supplier

SELECT \* FROM product

LEFT JOIN supplier ON supplier.id = product.supplier\_id

SELECT \* FROM product

LEFT JOIN department ON department.id = product.depatrment\_id

SELECT \* FROM product

RIGHT JOIN supplier ON supplier.id = product.supplier\_id

SELECT \* FROM product

RIGHT JOIN department ON department.id = product.depatrment\_id

SELECT \* FROM product

INNER JOIN supplier ON supplier.id = product.supplier\_id

SELECT \* FROM product

INNER JOIN department ON department.id = product.depatrment\_id