**Лабораторная работа № 5\_1**

## Основы SQL.

|  |
| --- |
| Настройка SQL\*Plus. Необходимо включить режим ECHO и вывести протокол лаб. работы в файл *<Фамилия студента>.txt*. Этот файл является отчетом о проделанной лаб. работе.  SQL> set echo on  SQL> spool z:\lab\spool\_5\_1.txt  ……………………………………..  Завершить протокол лаб. работы (команда spool off) и направить результаты преподавателю.  SQL> spool off |

**https://coderlessons.com/tutorials/bazy-dannykh/uchit-sql/sql-kratkoe-rukovodstvo**

## **ЗАДАНИЕ 1**

SQL> CREATE TABLE CUSTOMERS(

ID INT NOT NULL,

NAME VARCHAR (20) NOT NULL,

AGE INT NOT NULL,

ADDRESS CHAR (25) ,

SALARY DECIMAL (18, 2),

PRIMARY KEY (ID)

);

SQL> DESC CUSTOMERS;

INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)

VALUES (1, 'Ramesh', 32, 'Ahmedabad', 2000.00 );

INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)

VALUES (2, 'Khilan', 25, 'Delhi', 1500.00 );

INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)

VALUES (3, 'kaushik', 23, 'Kota', 2000.00 );

INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)

VALUES (4, 'Chaitali', 25, 'Mumbai', 6500.00 );

INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)

VALUES (5, 'Hardik', 27, 'Bhopal', 8500.00 );

INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)

VALUES (6, 'Komal', 22, 'MP', 4500.00 );

SQL>select \* from CUSTOMERS;

SQL>SELECT ID, NAME, SALARY FROM CUSTOMERS;

**ЗАДАНИЕ 2**

**Заполните одну таблицу, используя другую таблицу**

Вы можете заполнить данные в таблицу с помощью оператора выбора поверх другой таблицы; при условии, что другая таблица имеет набор полей, которые необходимы для заполнения первой таблицы.

Вот синтаксис –

INSERT INTO first\_table\_name [(column1, column2, ... columnN)]

SELECT column1, column2, ...columnN

FROM second\_table\_name

[WHERE condition];

**ЗАДАНИЕ 3**

**Выполнить блок coderlessons - SQL – выражения**

Выражение представляет собой комбинацию одного или нескольких значений, операторов и функций SQL, которые оценивают значение. Эти выражения SQL похожи на формулы и написаны на языке запросов. Вы также можете использовать их для запроса базы данных для определенного набора данных.

**SQL – Group By**

Предложение SQL **GROUP BY** используется в сотрудничестве с оператором SELECT для объединения идентичных данных в группы. Это предложение GROUP BY следует за предложением WHERE в инструкции SELECT и предшествует предложению ORDER BY.

**Синтаксис**

Основной синтаксис предложения GROUP BY показан в следующем блоке кода. Предложение GROUP BY должно соответствовать условиям в предложении WHERE и должно предшествовать предложению ORDER BY, если оно используется.

SELECT column1, column2

FROM table\_name

WHERE [ conditions ]

GROUP BY column1, column2

ORDER BY column1, column2

**пример**

Теперь давайте посмотрим на таблицу, в которой таблица CUSTOMERS содержит следующие записи с повторяющимися именами:

+----+----------+-----+-----------+----------+

| ID | NAME | AGE | ADDRESS | SALARY |

+----+----------+-----+-----------+----------+

| 1 | Ramesh | 32 | Ahmedabad | 2000.00 |

| 2 | Ramesh | 25 | Delhi | 1500.00 |

| 3 | kaushik | 23 | Kota | 2000.00 |

| 4 | kaushik | 25 | Mumbai | 6500.00 |

| 5 | Hardik | 27 | Bhopal | 8500.00 |

| 6 | Komal | 22 | MP | 4500.00 |

| 7 | Muffy | 24 | Indore | 10000.00 |

+----+----------+-----+-----------+----------+

Опять же, если вы хотите узнать общую сумму заработной платы по каждому клиенту, запрос GROUP BY будет выглядеть следующим образом:

SQL> SELECT NAME, SUM(SALARY) FROM CUSTOMERS

GROUP BY NAME;

Это даст следующий результат –

+---------+-------------+

| NAME | SUM(SALARY) |

+---------+-------------+

| Hardik | 8500.00 |

| kaushik | 8500.00 |

| Komal | 4500.00 |

| Muffy | 10000.00 |

| Ramesh | 3500.00 |

+---------+-------------+

**https://dzen.ru/a/Xz-sILfQ-yc08d-b**

