***Команды для работы с базами данных***

*1. Просмотр доступных баз данных*

SHOW DATABASES;

*2. Создание новой базы данных*

CREATE DATABASE;

*3. Выбор базы данных для использования*

USE <database\_name>;

*4. Удаление базы данных*

DROP DATABASE <database\_name>;

***Работа с таблицами***

*5. Просмотр таблиц, доступных в базе данных*

SHOW TABLES;

*6. Создание новой таблицы*

CREATE TABLE <table\_name1> (

<col\_name1> <col\_type1>,

<col\_name2> <col\_type2>,

<col\_name3> <col\_type3>

PRIMARY KEY (<col\_name1>),

FOREIGN KEY (<col\_name2>) REFERENCES <table\_name2>(<col\_name2>)

);

*7. Добавление данных в таблицу*

INSERT INTO <table\_name> (<col\_name1>, <col\_name2>, <col\_name3>, …)

VALUES (<value1>, <value2>, <value3>, …);

При добавлении данных в каждый столбец таблицы не требуется указывать названия столбцов.

INSERT INTO <table\_name>

VALUES (<value1>, <value2>, <value3>, …);

*8. Обновление данных таблицы*

UPDATE <table\_name>

SET <col\_name1> = <value1>, <col\_name2> = <value2>, ...

WHERE <condition>;

*9. Удаление всех данных из таблицы*

DELETE FROM <table\_name>;

*10. Удаление таблицы*

DROP TABLE <table\_name>;

***Команды для создания запросов***

*11. SELECT*

SELECT используется для получения данных из определённой таблицы:

SELECT <col\_name1>, <col\_name2>, …

FROM <table\_name>;

Следующей командой можно вывести все данные из таблицы:

SELECT \* FROM <table\_name>;

*12. WHERE*

Можно использовать ключевое слово WHERE в SELECT для указания условий в запросе:

SELECT <col\_name1>, <col\_name2>, …

FROM <table\_name>

WHERE <condition>;

Пример

SELECT \* FROM course WHERE dept\_name=’Comp. Sci.’;

SELECT \* FROM course WHERE credits>3;

SELECT \* FROM course WHERE dept\_name='Comp. Sci.' AND credits>3;

*13. GROUP BY*

Оператор GROUP BY часто используется с агрегатными функциями, такими как COUNT, MAX, MIN, SUM и AVG, для группировки выходных значений.

SELECT <col\_name1>, <col\_name2>, …

FROM <table\_name>

GROUP BY <col\_namex>;

Пример

Выведем количество курсов для каждого факультета:

SELECT COUNT(course\_id), dept\_name

FROM course

GROUP BY dept\_name;

*14. ORDER BY*

ORDER BY используется для сортировки результатов запроса по убыванию или возрастанию. ORDER BY отсортирует по возрастанию, если не будет указан способ сортировки ASC или DESC.

SELECT <col\_name1>, <col\_name2>, …

FROM <table\_name>

ORDER BY <col\_name1>, <col\_name2>, … ASC|DESC;

*15. BETWEEN*

BETWEEN используется для выбора значений данных из определённого промежутка. Могут быть использованы числовые и текстовые значения, а также даты.

SELECT <col\_name1>, <col\_name2>, …

FROM <table\_name>

WHERE <col\_namex> BETWEEN <value1> AND <value2>;

Пример

Выведем список инструкторов, чья зарплата больше 50 000, но меньше 100 000:

SELECT \* FROM instructor

WHERE salary BETWEEN 50000 AND 100000;

*16. LIKE*

Оператор LIKE используется в WHERE, чтобы задать шаблон поиска похожего значения.

Есть два свободных оператора, которые используются в LIKE:

* % (ни одного, один или несколько символов);
* \_ (один символ).

SELECT <col\_name1>, <col\_name2>, …

FROM <table\_name>

WHERE <col\_namex> LIKE <pattern>;

Пример

Выведем список курсов, в имени которых содержится «to», и список курсов, название которых начинается с «CS-»:

SELECT \* FROM course WHERE title LIKE ‘%to%’;

SELECT \* FROM course WHERE course\_id LIKE 'CS-\_\_\_';

*17. IN*

С помощью IN можно указать несколько значений для оператора WHERE:

SELECT <col\_name1>, <col\_name2>, …

FROM <table\_name>

WHERE <col\_namen> IN (<value1>, <value2>, …);

Пример

Выведем список студентов с направлений Comp. Sci., Physics и Elec. Eng.:

SELECT \* FROM student

WHERE dept\_name IN (‘Comp. Sci.’, ‘Physics’, ‘Elec. Eng.’);

*18. JOIN*

JOIN используется для связи двух или более таблиц с помощью общих атрибутов внутри них. На изображении ниже показаны различные способы объединения в SQL. Обратите внимание на разницу между левым внешним объединением и правым внешним объединением:



SELECT <col\_name1>, <col\_name2>, …

FROM <table\_name1>

JOIN <table\_name2>

ON <table\_name1.col\_namex> = <table2.col\_namex>;

Пример 1

Выведем список всех курсов и соответствующую информацию о факультетах:

SELECT \* FROM course

JOIN department

ON course.dept\_name=department.dept\_name;

Пример 2

Выведем список всех обязательных курсов и детали о них:

SELECT prereq.course\_id, title, dept\_name, credits, prereq\_id

FROM prereq

LEFT OUTER JOIN course

ON prereq.course\_id=course.course\_id;

Пример 3

Выведем список всех курсов вне зависимости от того, обязательны они или нет:

SELECT course.course\_id, title, dept\_name, credits, prereq\_id

FROM prereq

RIGHT OUTER JOIN course

ON prereq.course\_id=course.course\_id;

*19. Агрегатные функции*

Эти функции используются для получения совокупного результата, относящегося к рассматриваемым данным. Ниже приведены общеупотребительные агрегированные функции:

* COUNT (col\_name) — возвращает количество строк;
* SUM (col\_name) — возвращает сумму значений в данном столбце;
* AVG (col\_name) — возвращает среднее значение данного столбца;
* MIN (col\_name) — возвращает наименьшее значение данного столбца;
* MAX (col\_name) — возвращает наибольшее значение данного столбца.