Теория к практическим работам:

Практическая работа №1:

Синтаксис оператора SELECT для SQL1 имеет следующий вид:

*SELECT [ALL | DISTINCT] (<Список полей>|\*)*

*FROM <Список таблиц>*

*[WHERE <Предикат-условие селекции или соединения>]*

*[GROUP BY <Список полей группировки>]*

*[HAVING <Предикат-условие селекции для группы>]*

*[ORDER BY <Список полей, по которым упорядочивается вывод записей>]*

LIKE может принимать аргумент:

LIKE ‘M%’ – все значения, которые начинаются на «М».

LIKE ‘%M’ – все значения, которые заканчиваются на «М».

LIKE ‘\_M%’ – все значения, у которых вторая буква «М».

LIKE ‘%M%’ – все значения, у которых есть буква «М».

Практическая работа №2:

Запросы могут производить обобщенное групповое значение полей точно также, как и значение одного пол. Это делает с помощью агрегатных функций. Агрегатные функции производят одиночное значение для всей группы таблицы. Имеется список этих функций:

COUNT - производит номера строк или не-NULL значения полей, которые выбрал.

SUM - производит арифметическую сумму всех выбранных значений.

AVG - производит усреднение всех выбранных значений.

MAX - производит наибольшее из всех выбранных значений.

MIN - производит наименьшее из всех выбранных значений.

Групповые функции используются подобно именам полей в операторе SELECT, но с одним исключением: они берут имя поля как аргумент.

С функциями SUM и AVG могут использоваться только числовые поля.

С функциями COUNT, MAX и MIN могут использоваться как числовые, так и символьные поля. При использова­нии с символьными полями МАХ и MIN будут транслировать их в эквивалент ASCII кода и обрабатывать в алфавитном порядке.

Если предложение WHERE определяет предикат для фильтрации строк, то предложение HAVING применяется после группировки для определения аналогичного предиката, фильтрующего группы по значениям агрегатных функций. Это предложение необходимо для проверки значений, которые получены с помощью агрегатной функции не из отдельных строк источника записей, определенного в предложении FROM, а из групп таких строк. Поэтому такая проверка не может содержаться в предложении WHERE.

Оператор SQL ORDER BY выполняет сортировку выходных значений. Оператор SQL ORDER BY можно применять как к числовым столбцам, так и к строковым. В последнем случае, сортировка будет происходить по алфавиту.

Оператор SQL ORDER BY имеет следующий синтаксис:

|  |  |
| --- | --- |
|  | ORDER BY column\_name [ASC | DESC] |

Сортировка может производиться как по возрастанию, так и по убыванию значений.

Параметр ASC (по умолчанию) устанавливает порядок сортирования возрастанию, от меньших значений к большим.

Параметр DECS устанавливает порядок сортирования по убыванию, от больших значений к меньшим.

Практическая работа №3:

Бывают вида действий над таблицами:

1. INSERT (заполнение таблицы)
2. DELETE (удаление таблицы)
3. UPDATE (обновление таблицы)
4. CREATE TABLE (добавление таблицы)

Практическая работа №4:

Оператор SQL INNER JOIN формирует таблицу из записей двух или нескольких таблиц. Каждая строка из первой (левой) таблицы, сопоставляется с каждой строкой из второй (правой) таблицы, после чего происходит проверка условия. Если условие истинно, то строки попадают в результирующую таблицу. В результирующей таблице строки формируются конкатенацией строк первой и второй таблиц.

Левое внешнее объединение (LEFT OUTER JOIN). В этом случае получаем все записи, удовлетворяющие условию объединения, плюс все оставшиеся записи из внешней таблицы, которые не удовлетворяют условию объединения.

То есть мы получили все записи, которые есть в обоих таблицах. Записи, у которых значение поля id совпадает соединяются, а у записей, для которых совпадений не найдено недостающие поля заполняются значением NULL.

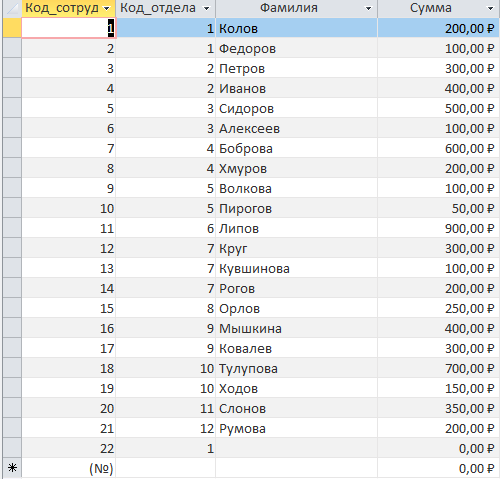
Правое внешнее объединение (RIGHT OUTER JOIN). Оно работает точно также, как и левое объединение, только в качестве внешней таблицы будет использоваться правая (в нашем случае таблица Б на графе).

Практическая работа №5:

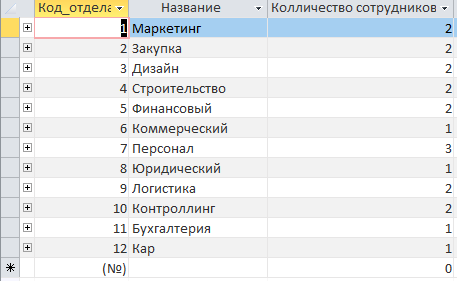
Вложенный запрос – это запрос, который находится внутри другого SQL запроса и встроен внутри условного оператора WHERE.  
Данный вид запросов используется для возвращения данных, которые будут использоваться в основном запросе, как условие для ограничения получаемых данных.  
Вложенные запросы должны следовать следующим правилам:

* Вложенный запрос должен быть заключён в родительский запрос.
* Вложенный запрос может содержать только одну колонку в операторе SELECT.
* Оператор ORDER BY не может быть использован во вложенном запросе. Для обеспечения функционала ORDER BY, во вложенном запросе может быть использован GROUP BY.
* Вложенные запросы, возвращающие более одной записи могут использоваться с операторами нескольких значений, как оператор IN.
* Вложенный запрос не может заканчиваться в функции.
* SELECT не может включать никаких ссылок на значения BLOB, ARRAY, CLOB и NCLOB.
* Оператор BETWEEN не может быть использован вместе с вложенным запросом.

Изначальные таблицы:



Сотрудники



Отделы