**Концептуальная схема выполнения оператора SELECT. Синтаксис оператора выборки данных. Синтаксис соединенных таблиц. Синтаксис условных выражений.**

Концептуальная схема выполнения оператора SELECT.

1. вычисляется прямое декартово произведение всех таблиц, указанных в обязательном разделе FROM, получаем некоторую таблицу A.
2. если присутствует WHERE, то сканируется таблица A, для каждой строки таблицы A вычисляется условие из этого раздела; в результат включаются строки, для которых условие возвращает значение TRUE, результат: таблица B, если WHERE нет, то шаг 3
3. если присутствует GROUP BY, то строки таблицы B группируются в соответствии со списком группировки из раздела, получаем таблицу С. Если нет, то шаг 4.
4. если присутствует HAVING, то группы, не удовлетворяющие условию из этого раздела исключаются, в результате таблица D
5. каждая группа, полученная на шаге 4, генерирует одну строку результата следующим образом: вычисляются все скалярные выражения, указанные в разделе SELECT для каждой группы, вычисляются агрегатные функции, приведенные в разделе SELECT. Если GROUP BY нет, а функции есть, то считается, что имеется одна группа. Если нет агрегатных функций, ни раздела, то считается, что есть столько групп, сколько строк. В результате таблица E содержит столько колонок, сколько элементов приведено в разделе SELECT и столько строк, сколько отобрано групп. Если есть ORDER BY, то строки таблицы упорядочиваются в соответствии со списком упорядочивания.

Синтаксис оператора выборки данных:

Табличное выражение:

[ORDER BY {{имя столбца результата [ASC|DESC|}

|{положит целое [ASC|DESC}]

табл выражение := select выражение

[UNION | EXCEPT | INTERSECT | ALL }

{SELECT выражение | TABLE имя табл | конструктор значений таблицы}

SELECT выр-е:= SELECT [ALL|DISTINCT]

{{скалярное выражение | функция агрегирования |

SELECT выр-е} [AS имя столбца]}…}

|{имя табл | имя корреляции}…}

FROM {имя табл [AS] [имя корреляции]

[(имя столбца,…)]}

|SELECT выр-я [AS] имя корреляции

[(имя столбца,…)]}

| соединенная таблица}…

[WHERE условное выражение]

[GROUP BY [имя табл|имя корреляции}]

имя столбца]…}

[HAVING условное выражение];

Условное выражение вычисляется для каждой строки, являющейся кандидатом в результирующее множество строк. Можно использовать подзапросы. Раздел HAVING содержит условное выражение, вычисляющееся для каждой группы в соответствии со списком группировки в разделе GROUP BY. Если в разделе SELECT присутствует агрегатная функция и присутствует раздел GROUP BY, то агрегатная функция вычисляется для каждой группы отдельно. Если раздел отсутствует, то агрегатная функция вычисляется по всем строкам, удовлетворяющим условному выражению в разделе WHERE. Скалярное выражение – имя столбцов таблицы или функции.

Синтаксис соединенных таблиц:

Соединенная таблица :=

* перекрестное соединение - A CROSS JOIN B - декартово произведение таблицы
* естественное соединение - A NATURAL [тип соединения] JOIN B - произведение по всем столбцам таблиц, имеющим одинаковые имена.
* соединение посредством предиката - A [тип соединения] JOIN B ON предикат - соединение по общим столбцам - BETWEEN | IN | LIKE | IS NULL | EXIST | UNIQUE | MATCH | OVERLAPS
* соединение посредством имен столбцов - A [тип соед-я] JOIN B USING {имя столбца}
* соединение объединения - A UNION JOIN B – обратное к внутреннему соединению, обратно FULL

Тип соединения

INNER - соединяются те строки, для которых найдены совпадения

| LEFT [OUTER] – результат включает все строки из левой таблицы и те из правой таблицы, для которых найдены совпадения. Для строк из A, для которых не найдены совпадения в B, заносятся со значениями NULL.

| RIGHT [OUTER]

| FULL [OUTER] – все строки таблиц, для совпадающих – реальные значения, для несовпадающих – NULL.

Синтаксис условных выражений раздела WHERE:

Условное выражение:= [( ] [NOT]

{Предикат сравнения | BETWEEN | IN | LIKE | NULL | предикат количественного сравнения | EXIST | UNIQUE | MATСH | OVERLAPS | } [ { AND | OR } условное выражение] [ ) ] [IS [NOT] {TRUE | FALSE | UNKNOWN}

BETWEEN – диапазоны значения

IN – проверяет вхождение во множество

EXIST – проверяет существует ли предикат подзапроса

MATСH – проверяет будет ли значение в любой строке совпадать с результатом подзапроса.