

порядке а) убывания семестров и б) возрастания отводимых на предмет часов. Поле семестра в выходных данных должно быть первым, за ним должны следовать имя предмета обучения и идентификатор предмета.

24. Напишите запрос, который выполняет вывод суммы баллов всех студентов для каждой даты сдачи экзаменов и представляет результаты в порядке убывания этих сумм.

25. Напишите запрос, который выполняет вывод а) среднего, б) минимального, в) максимального баллов всех студентов для каждой даты сдачи экзаменов, и представляет результаты в порядке убывания этих значений.

2.8. Вложенные подзапросы

SQL позволяет использовать одни запросы внутри других запросов, то есть вкладывать запросы друг в друга. Предположим, известна фамилия студента (“Петров”), но неизвестно значение поля `STUDENT_ID` для него. Чтобы извлечь данные обо всех оценках этого студента, можно записать следующий запрос:

```
SELECT *
FROM EXAM_MARKS
WHERE STUDENT_ID =
    ( SELECT STUDENT_ID
      FROM STUDENT SURNAME = ‘Петров’ );
```

Как работает запрос SQL со связанным подзапросом?

- Выбирается строка из таблицы, имя которой указано во внешнем запросе.
- Выполняется подзапрос и полученное в результате его выполнения значение применяется для анализа этой строки в условии предложения **WHERE** внешнего запроса.
- По результату оценки этого условия принимается решение о включении или не включении строки в состав выходных данных.
- Процедура повторяется для следующей строки таблицы внешнего запроса.

Следует обратить внимание, что приведенный выше запрос корректен только в том случае, если в результате выполнения указанного в скобках подзапроса возвращается *единственное значение*. Если в результате выполнения подзапроса будет возвращено несколько значений, то этот подзапрос будет ошибочным. В данном примере это произойдет, если в таблице STUDENT будет несколько записей со значениями поля SURNAME = 'Петров'.

В некоторых случаях для гарантии получения единственного значения в результате выполнения подзапроса используется **DISTINCT**. Одним из видов функций, которые автоматически *всегда* выдают в результате единственное значение для любого количества строк, являются агрегирующие функции.

Оператор **IN** также широко применяется в подзапросах. Он задает список значений, с которыми сравниваются другие значения для определения истинности задаваемого этим оператором предиката.

Данные обо всех оценках (таблица EXAM_MARKS) студентов из Воронежа можно выбрать с помощью следующего запроса:

```
SELECT *
FROM EXAM_MARKS
WHERE STUDENT_ID IN
  ( SELECT STUDENT_ID
    FROM STUDENT
    WHERE CITY = 'Воронеж');
```

Подзапросы можно применять внутри предложения **HAVING**. Пусть требуется определить количество предметов обучения с оценкой, превышающей среднее значение оценки студента с идентификатором 301:

```
SELECT COUNT(DISTINCT SUBJ_ID), MARK
FROM EXAM_MARKS
GROUP BY MARK
HAVING MARK >
  ( SELECT AVG(MARK)
    FROM EXAM_MARKS
    WHERE STUDENT_ID = 301);
```

2.9. Формирование связанных подзапросов

При использовании подзапросов во внутреннем запросе можно ссылаться на таблицу, имя которой указано в предложении **FROM** внешнего запроса. В этом случае такой *связанный* подзапрос выполняется по одному разу для *каждой* строки таблицы основного запроса.

Пример: выбрать сведения обо всех предметах обучения, по которым проводился экзамен 20 января 1999 г.

```
SELECT *
FROM SUBJECT SU
WHERE '20/01/1999' IN
  ( SELECT EXAM_DATE
    FROM EXAM_MARKS EX
    WHERE SU.SUBJ_ID = EX.SUBJ_ID);
```

В некоторых СУБД для выполнения этого запроса, возможно, потребуется преобразование значения даты в символьный тип. В приведенном запросе SU и EX являются псевдонимами (алиасами), то есть специально вводимыми именами, которые могут быть использованы в данном запросе вместо настоящих имен. В приведенном примере они используются вместо имен таблиц SUBJECT и EXAM_MARKS. .

Эту же задачу можно решить с помощью операции соединения таблиц:

```
SELECT DISTINCT SU.SUBJ_ID, SUBJ_NAME, HOUR, SEMESTER
FROM SUBJECT FIRST, EXAM_MARKS SECOND
WHERE FIRST.SUBJ_ID = SECOND.SUBJ_ID
AND SECOND.EXAM_DATE = '20/01/1999';
```

В этом выражении алиасами таблиц являются имена FIRST и SECOND.

Можно использовать подзапросы, связывающие таблицу со своей собственной копией. Например, надо найти идентификаторы, фамилии и стипендии студентов, получающих стипендию выше средней на курсе, на котором они учатся.

```
SELECT DISTINCT STUDENT_ID, SURNAME, STIPEND
FROM STUDENT E1
WHERE STIPEND >
  ( SELECT AVG(STIPEND)
    FROM STUDENT E2
```

```
WHERE E1.KURS = E2.KURS);
```

Тот же результат можно получить с помощью следующего запроса:

```
SELECT DISTINCT STUDENT_ID, SURNAME, STIPEND
FROM STUDENT E1,
  (SELECT KURS, AVG(STIPEND) AS AVG_STIPEND
   FROM STUDENT E2
   GROUP BY E2.KURS) E3
WHERE E1.STIPEND > AVG_STIPEND AND E1.KURS=E3.KURS;
```

Обратите внимание – второй запрос будет выполнен гораздо быстрее. Дело в том, что в первом варианте запроса агрегирующая функция **AVG** выполняется над таблицей, указанной в подзапросе, для *каждой* строки внешнего запроса. В другом варианте вторая таблица (алиас E2) обрабатывается агрегирующей функцией один раз, в результате чего формируется вспомогательная таблица (в запросе она имеет алиас E3), со строками которой затем соединяются строки первой таблицы (алиас E1). Следует иметь в виду, что реальное время выполнения запроса в большой степени зависит от оптимизатора запросов конкретной СУБД.

2.10. Связанные подзапросы в HAVING

В разделе 2.4 указывалось, что предложение **GROUP BY** позволяет группировать выводимые **SELECT**-запросом записи по значению некоторого поля. Использование предложения **HAVING** позволяет при выводе осуществлять фильтрацию таких групп. Предикат предложения **HAVING** оценивается не для каждой строки результата, а для каждой группы выходных записей, сформированной предложением **GROUP BY** внешнего запроса.

Пусть, например, необходимо по данным из таблицы EXAM_MARKS определить сумму полученных студентами оценок (значений поля MARK), сгруппировав значения оценок по датам экзаменов и исключив те дни, когда число студентов, сдававших в течение дня экзамены, было меньше 10.

```
SELECT EXAM_DATE, SUM(MARK)
FROM EXAM_MARKS A
GROUP BY EXAM_DATE
```

```

HAVING 10 <
( SELECT COUNT(MARK)
  FROM EXAM_MARKS B
 WHERE A.EXAM_DATE = B.EXAM_DATE);

```

Подзапрос вычисляет количество строк с одной и той же датой, совпадающей с датой, для которой сформирована очередная группа основного запроса.

УПРАЖНЕНИЯ

26. Напишите запрос с подзапросом для получения данных обо всех оценках студента с фамилией “Иванов”. Предположим, что его персональный номер не известен. Всегда ли такой запрос будет корректным?
27. Напишите запрос, выбирающий данные об именах всех студентов, имеющих по предмету с идентификатором 101 балл выше общего среднего балла.
28. Напишите запрос, который выполняет выборку имен всех студентов, имеющих по предмету с идентификатором 102 балл ниже общего среднего балла.
29. Напишите запрос, выполняющий вывод количества предметов, по которым экзаменовался каждый студент, сдававший более 20-ти предметов.
30. Напишите команду **SELECT**, использующую связанные подзапросы и выполняющую вывод имен и идентификаторов студентов, у которых стипендия совпадает с максимальным значением стипендии для города, в котором живет студент.
31. Напишите запрос, который позволяет вывести имена и идентификаторы всех студентов, для которых точно известно, что они проживают в городе, где нет ни одного университета.
32. Напишите два запроса, которые позволяют вывести имена и идентификаторы всех студентов, для которых точно известно, что они проживают не в том городе, где расположен их университет. Один запрос с использованием соединения, а другой – с использованием связанного

подзапроса.

2.11. Использование оператора **EXISTS**

Используемый в SQL оператор **EXISTS** (СУЩЕСТВУЕТ) генерирует значение **ИСТИНА** или **ЛОЖЬ**, подобно булеву выражению. Используя подзапросы в качестве аргумента, этот оператор оценивает результат выполнения подзапроса как истинный, если этот подзапрос генерирует выходные данные, то есть в случае *существования* (возврата) хотя бы одного найденного значения. В противном случае результат подзапроса – ложный. Оператор **EXISTS** не может принимать значение **unknown** (неизвестно).

Пусть, например, нужно извлечь из таблицы EXAM_MARKS данные о студентах, получивших хотя бы одну неудовлетворительную оценку.

```
SELECT DISTINCT STUDENT_ID
FROM EXAM_MARKS A
WHERE EXISTS
  (SELECT *
   FROM EXAM_MARKS B
   WHERE MARK < 3
   AND B.STUDENT_ID=A.STUDENT_ID);
```

При использовании связанных подзапросов предложение **EXISTS** анализирует каждую строку таблицы, на которую имеется ссылка во внешнем запросе. Главный запрос получает строки-кандидаты на проверку условия. Для каждой строки-кандидата выполняется подзапрос. Как только подзапрос находит строку, где в столбце MARK значение удовлетворяет условию, он прекращает выполнение и возвращает значение **ИСТИНА** внешнему запросу, который затем анализирует свою строку-кандидата.

Например, требуется получить идентификаторы предметов обучения, экзамены по которым сдавались не одним, а несколькими студентами: