



Язык SQL

(Structured Query Language)

Алфавит языка

Буквы: A...Z, a...z;

Цифры: 0..9;

Символы: + - * / ! @ \$ = < > ^ ' () | _ ; , .

Идентификаторы

30 символов

Буквы

Цифры

\$

#

—

Комментарии

Однострочные

-- текст комментария

Многострочные

/* текст комментария */

Литералы

CHAR ‘ ’

NUMBER

Пустые значения.

NULL

Псевдостолбцы.

ROWID

ROWNUM

LEVEL

CURRVAL

имя_последовательности. CURRVAL

NEXTVAL

имя_последовательности. NEXTVAL

Типы данных языка SQL

- Символьные типы

CHAR(длина) до 2000 байт

VARCHAR2(длина) до 4000 байт

LONG до 2 Гб

- Числовые типы

NUMBER от $1,0 \cdot 10^{-130}$ до $1,0 \cdot 10^{126} - 1$

NUMBER(p) p от 1 до 38

NUMBER(p,s) p от 1 до 38, s от -84 до 127

Типы данных стандарта ANSI SQL: INTEGER, SMALLINT, DECIMAL, FLOAT, REAL и др.

- Тип данных DATE

от 1 января 4712 г. до н.э. до 31 декабря 9999 г. н.э.

SYSDATE

'DD-MM-YY'

Типы данных языка SQL

- Двоичные типы данных

RAW(длина) до 2000 байт

LONGRAW до 2 Гб

- Большие объекты (LOB-объекты)

CLOB

BLOB

BFILE

Операторы языка SQL

- 1) Операторы языка описания данных – DDL (Data Definition Language)

CREATE, ALTER, DROP, GRANT, REVOKE

- 2) Операторы языка манипулирования данными – DML (Data Manipulation Language)

INSERT, DELETE, UPDATE, SELECT

- 3) Операторы управления транзакциями

COMMIT, ROLLBACK

- 4) Операторы управления сеансом

ALTER SESSION, SET ROLE

- 5) Операторы управления системой

ALTER SYSTEM

Операции языка SQL

- Арифметические операции

1. Унарные: +, -
2. Бинарные: +, -, *, /

- Операции над строками

Сцепление строк ||

- Операции сравнения

=

!= <> ^=

< > <= >=

операнд BETWEEN нач_значение AND кон_значение

операнд IN (список выражений | подзапрос)

операнд NOT IN (список выражений | подзапрос)

Операции языка SQL

операнд LIKE шаблон % _

операнд IS [NOT] NULL

операция сравнения с квантором ANY

операция сравнения с квантором ALL

операция сравнения EXISTS

- Логические операции

NOT, AND, OR

- Операции над множествами

UNION ALL, UNION, INTERSECT, MINUS

Функции языка SQL

- Числовые функции

ABS (n)

ROUND (n, [r])

MOD (m,n)

POWER (m,n)

SQRT (n)

- Символьные функции

UPPER (str)

LOWER (str)

LENGTH (str)

INITCAP (str)

SUBSTR (str,n,m)

LPAD (str,n,chr)

RPAD (str,n,chr)

- Функции преобразования типа
TO_CHAR (d1,[fmt])
TO_NUMBER (char,[fmt]))
TO_DATE (char,[fmt])
- Групповые функции
COUNT (*)
COUNT ([DISTINCT] выражение)
SUM ([DISTINCT] выражение)
AVG ([DISTINCT] выражение)
MAX ([DISTINCT] выражение)
MIN ([DISTINCT] выражение)

- *NVL*

NVL (выражение1, выражение2)

NVL2 (выражение1, выражение2, выражение3)

NULLIF (выражение1, выражение2)

COALESCE (выражение1, выражение2, ...,
выражениеn)

Создание таблиц

```
CREATE TABLE имя_таблицы      <определение_поля>:  
(<определение_поля>          имя_поля тип_данных  
[,<определение_поля>...]     [DEFAULT значение]  
[,<ограничение_таблицы>]...) | [NULL | NOT NULL]  
AS подзапрос;                [<ограничение_поля>]
```

Пример 1

```
CREATE TABLE EMP  
  (EMPNO NUMBER(4) PRIMARY KEY,  
   EMPNAME VARCHAR2(30) NOT NULL,  
   BIRTHDATE DATE);
```

Ограничения поля

[CONSTRAINT имя_ограничения>] тип_ограничения

- PRIMARY KEY
- UNIQUE
- NOT NULL
- CHECK (условие)
- REFERENCES

[CONSTRAINT <имя_ограничения>]

REFERENCES имя_таблицы [(имя_поля)]

[ON DELETE CASCADE| ON DELETE SET NULL]

Пример 2

```
CREATE TABLE SALARY
```

```
    (EMPNO NUMBER(4) REFERENCES EMP(EMPNO),  
     MONTH NUMBER(2) CHECK(MONTH>0 AND  
MONTH<13),  
     YEAR NUMBER(4) CHECK(YEAR>1987 AND  
YEAR<2011),  
     SALVALUE NUMBER(6));
```

Ограничения таблицы

[CONSTRAINT имя_ограничения] тип_ограничения

- PRIMARY KEY

[CONSTRAINT <имя_ограничения>]

PRIMARY KEY (имя_поля [,<имя_поля>...])

- UNIQUE

[CONSTRAINT <имя_ограничения>] UNIQUE (имя_поля
[,<имя_поля>...])

- CHECK (условие)

- REFERENCES

[CONSTRAINT <имя_ограничения>]

FOREIGN KEY (имя_поля [,<имя_поля>...])

REFERENCES имя_таблицы (имя_поля [,<имя_поля>...])

[ON DELETE CASCADE| ON DELETE SET NULL]

Создание последовательности

```
CREATE SEQUENCE имя_последовательности  
[START WITH начальное_значение]  
[INCREMENT BY шаг];
```

имя_последовательности.NEXTVAL

Пример

```
CREATE SEQUENCE sq_emp START WITH 1001 MAXVALUE 9999;
```

```
CREATE TABLE EMP (  
    EMPNO NUMBER(4) PRIMARY KEY DEFAULT sq_emp.nextval,  
    ...);
```

Изменение структуры таблицы

- добавление поля в таблицу

```
ALTER TABLE имя_таблицы ADD (<определение_поля>  
[,<определение_поля>...]);
```

- изменение определения поля

```
ALTER TABLE имя_таблицы MODIFY [COLUMN]  
(<определение_поля>  
[,<определение_поля>...]);
```

- добавление нового ограничения

```
ALTER TABLE имя_таблицы ADD CONSTRAINT  
<определение_ограничения>;
```

- удаление первичного ключа таблицы

```
ALTER TABLE имя_таблицы DROP PRIMARY KEY;
```

- переименование поля

```
ALTER TABLE имя_таблицы  
RENAME COLUMN старое_имя_поля TO новое_имя_поля;
```

Пример 3

```
ALTER TABLE emp
  ADD (sex CHAR(1)
       CHECK(sex IN('m', 'f')),
       marriage CHAR(1)
       DEFAULT 's' CHECK(marriage IN('s', 'm', 'd', 'w')));
```

Удаление таблицы

```
DROP TABLE имя_таблицы [<CASCADE CONSTRAINTS>];
```

Пример 4

```
DROP TABLE emp CASCADE CONSTRAINTS ;
```

Пример 5

```
ALTER TABLE emp DROP (marriage);
```

Индексы – это объекты базы данных, созданные для ускорения поиска данных в определенной таблице.

Создание индекса

```
CREATE [UNIQUE] INDEX имя_индекса ON имя_таблицы  
(имя_поля [ASC | DESC] [,имя_поля [ASC | DESC] ...]);
```

Базовый запрос SELECT

```
SELECT [DISTINCT] описание_результата  
FROM источник;
```

описание_результата

имя_поля_1 [AS имя] <, имя_поля_2 [AS имя], ...> | *

Упорядочивание строк

ORDER BY *<имястолбца | номерстолбца* [ASC | DESC] >...

Пример 6

```
SELECT EMPNAME, BIRTHDATE  
FROM emp  
ORDER BY BIRTHDATE DESC;
```

Условие выбора строк

WHERE условие_выбора

Пример 7

```
SELECT AVG(salvalue)
FROM salary WHERE month = 1 AND year = 2003;
```

Пример 8

```
SELECT DEPTNAME, DEPTADDR
FROM DEPT
WHERE DEPTNAME = '&DEPARTMENT';
```


Группирование строк

GROUP BY список_группировочных_столбцов
[HAVING *условие_отбора_в_группу*]

Пример 15

Найти среднюю начисленную зарплату за 2002 год в разрезе работников. Включать в результат только тех работников, начисления которым проводились не менее двух раз.

```
SELECT empno, AVG(salvalue) FROM salary  
WHERE year = 2002  
GROUP BY empno  
HAVING count(*)>1;
```

Группирование строк

Пример 16

Найти среднюю начисленную зарплату за 2002 год в разрезе работников. При расчете исключить тех работников, для которых максимальная начисленная зарплата была меньше 100.

```
SELECT empno, AVG(salvalue)
FROM salary s1
  WHERE year = 2002
  GROUP BY empno
  HAVING 100 <
(SELECT MAX(salvalue) FROM salary s2
  WHERE s1.empno=s2.empno);
```

Группирование строк

Пример 17

Вывести информацию о средней заработной плате (округлить с точностью до 2 знаков после запятой) по каждому месяцу каждого года. Произвести сортировку по году.

```
SELECT YEAR,MONTH, ROUND(AVG(SALVALUE),2) AS  
    AVG_SALARY  
FROM SALARY  
GROUP BY YEAR,MONTH  
ORDER BY YEAR;
```

Подзапросы (вложенные запросы)

- Некоррелированные

Пример 9

Получить имя самого старшего работника, а также его возраст (в днях).

```
SELECT empname, SYSDATE-birthdate FROM emp
WHERE birthdate =
  (SELECT MIN(birthdate) FROM emp);
```

```
SELECT e1.empname, SYSDATE-birthdate FROM emp e1
WHERE birthdate =
  (SELECT MIN(birthdate) FROM emp e2);
```

Подзапросы

- Коррелированные

Пример 10

Найти фамилии работников, которым не была начислена зарплата в январе 2003 года.

```
SELECT empname FROM emp WHERE NOT EXISTS(  
  SELECT * FROM salary  
  WHERE emp.empno = salary.empno AND month = 1  
  AND year = 2003);
```

Подзапросы

Пример 11

Получить информацию о кодах, названиях отделов и количестве работающих в этих отделах в настоящее время сотрудников.

```
SELECT deptno, deptname,  
       (SELECT COUNT(jobno) FROM career  
        WHERE career.deptno=dept.deptno AND enddate IS NULL)  
FROM dept;
```

Подзапросы

Пример 12

Найти коды работников, зарплата которых в ноябре 2007 года снизилась по сравнению с каким-либо предыдущим месяцем этого же года.

```
SELECT EMPNO FROM SALARY S1
WHERE MONTH = 11 AND YEAR = 2007
AND SALVALUE < ANY(
  SELECT SALVALUE FROM SALARY S2
  WHERE S1. EMPNO = S2. EMPNO AND S2.MONTH < 11
  AND S2.YEAR = 2007);
```

Подзапросы

Пример 13

Найти коды работников, зарплата которых в ноябре 2007 года возросла по сравнению с каким-либо предыдущим месяцем этого же года.

```
SELECT EMPNO FROM SALARY S1
WHERE MONTH = 11 AND YEAR = 2007
AND SALVALUE > ALL(
  SELECT SALVALUE FROM SALARY S2
  WHERE S1.EMPNO = S2.EMPNO AND S2.MONTH < 11
  AND S2.YEAR = 2007);
```


Встроенные представления

Пример 14

Выдать максимальную среднюю зарплату сотрудника

```
SELECT MAX (AVERAGE_SALEVALUE) AS  
MAX_AVERAGE_SALEVALUE  
FROM  
    (SELECT EMPNO, AVG(SALVALUE) AS  
AVERAGE_SALEVALUE  
    FROM SALARY  
    GROUP BY EMPNO);
```

Соединение

- Эквисоединение
- Декартово произведение таблиц
- Самосоединение
- Внешнее соединение
- Рефлексивные соединения

Соединение

Пример 18

Вывести имена сотрудников проработавших менее 1 года.

Вариант 1. WHERE

```
SELECT DISTINCT EMPNAME
```

```
FROM EMP E, CAREER C
```

```
WHERE E.EMPNO=C.EMPNO
```

```
AND MONTHS_BETWEEN(STARTDATE,ENDDATE)<12;
```

Соединение

отношение1 [INNER] JOIN отношение2 ON
условие_соединения

Вариант 2. JOIN ON

```
SELECT DISTINCT EMPNAME
```

```
FROM EMP E JOIN CAREER C ON E.EMPNO=C.EMPNO
```

```
WHERE MONTHS_BETWEEN(STARTDATE,ENDDATE)<12;
```

Соединение

Вариант 3. JOIN USING

```
SELECT DISTINCT EMPNAME  
FROM EMP JOIN CAREER USING (EMPNO)  
WHERE MONTHS_BETWEEN(STARTDATE,ENDDATE)<12;
```

Левое соединение

отношение1 LEFT OUTER JOIN отношение2 ON
усл_соединения

Пример 19

Найти имена работников и стаж их непрерывной работы (на одной должности и в одном отделе). Включить информацию о работниках, для которых по какой-то причине нет записей в таблице CAREER.

Вариант 1

```
SELECT EMPNAME, NVL(ENDDATE,  
SYSDATE)-STARTDATE AS STAGE  
FROM EMP LEFT OUTER JOIN CAREER ON EMP.EMPNO  
=CAREER.EMPNO;
```

Левое соединение

Пример 19

Вариант 2

```
SELECT EMPNAME, NVL(ENDDATE, SYSDATE)-STARTDATE AS  
STAGE  
FROM EMP, CAREER  
WHERE EMP.EMPNO = CAREER.EMPNO(+);
```

Правое соединение

отношение1 RIGHT OUTER JOIN отношение2 ON
усл_соединения

Пример 19

Вариант 3

```
SELECT EMPNAME, NVL(ENDDATE, SYSDATE)-STARTDATE AS  
STAGE  
FROM CAREER RIGHT OUTER JOIN EMP  
ON CAREER.EMPNO= EMP.EMPNO;
```


Правое соединение

Пример 19

Вариант 4

```
SELECT EMPNAME, NVL(ENDDATE, SYSDATE)-STARTDATE AS  
STAGE  
FROM EMP, CAREER  
WHERE CAREER.EMPNO(+) = EMP.EMPNO;
```

Соединение

Пример 20

Вывести названия отделов, дату начала и дату окончания работы сотрудника в отделе; чтобы задать фамилию сотрудника использовать переменную подстановки.

```
SELECT DEPTNAME,STARTDATE,ENDDATE  
FROM EMP JOIN (CAREER JOIN DEPT USING (DEPTNO))  
USING (EMPNO)  
WHERE EMPNAME='&EMPLOYEE_NAME';
```

Соединение

Пример 21

Вывести значение наибольшей зарплаты полученной сотрудником по имени SMITH с указанием месяца и года получения.

```
SELECT E.EMPNAME, S.SALVALUE, S.MONTH, S.YEAR
FROM SALARY S JOIN EMP E ON S.EMPNO=E.EMPNO
WHERE SALVALUE =
(SELECT MAX(S1.SALVALUE)
FROM SALARY S1 JOIN EMP E1 ON S1.EMPNO=E1.EMPNO
WHERE E1.EMPNAME='SMITH') AND E.EMPNAME='SMITH');
```

Группирование строк Соединение

Пример 22

Вывести номер сотрудника с минимальной заработной платой в каждом году.

Вариант 1

```
SELECT DISTINCT EMPNO,SALVALUE,YEAR
FROM SALARY
WHERE (SALVALUE,YEAR) IN
(SELECT MIN(SALVALUE),YEAR
FROM SALARY
GROUP BY YEAR);
```

Соединение

Пример 22

Вывести номер сотрудника с минимальной заработной платой в каждом году.

Вариант 2

```
SELECT DISTINCT EMPNO,SALVALUE,YEAR
FROM SALARY S1
WHERE SALVALUE=
(SELECT MIN(SALVALUE)
FROM SALARY S2
WHERE S1.YEAR=S2.YEAR);
```

Соединение

Пример 23

Вывести сотрудников, которые не занимали должность финансовый директор (FINANCIAL DIRECTOR) или должность исполнительный директор (EXECUTIVE DIRECTOR)

```
SELECT DISTINCT EMPNAME FROM EMP
WHERE NOT EXISTS (
SELECT * FROM CAREER JOIN JOB USING (JOBNO)
WHERE JOBNAME LIKE '%DIRECTOR' AND
EMP.EMPNO = CAREER.EMPNO);
```

Рефлексивное соединение

Таблица JOB с добавленным столбцом SUPERVISER_JOBNO

JOBNO	JOBNAME	MINSALARY	SUPERVISER_JOBNO
1000	MANAGER	2500	1002
1001	FINANCIAL DIRECTOR	7500	1006
1002	EXECUTIVE DIRECTOR	8000	1006
1003	SALESMAN	1500	1000
1004	CLERK	500	1000
1005	DRIVER	1800	1002
1006	PRESIDENT	15000	

Рефлексивное соединение

Пример 24

Вывести номер должности, зарплату соответствующую этой должности и зарплату соответствующей руководящей должности

```
SELECT a.JOBNO, a.MINSALARY AS JOB_SALARY,  
b.MINSALARY AS SUPERVISER_SALARY  
FROM JOB a JOIN JOB b  
ON a.SUPERVISER_JOBNO = b.JOBNO;
```

JOBNO	JOB_SALARY	SUPERVISER_SALARY
1004	500	2500
1003	1500	2500
1005	1800	8000
1000	2500	8000
1002	8000	15000
1001	7500	15000

Рефлексивное соединение

Пример 25

Вывести код сотрудника, имя сотрудника и разницу в зарплате сотрудника и его руководителя

```
SELECT a.JOBNO, d.EMPNAME,  
b.MINSALARY - a.MINSALARY AS SALARY_DIFF,  
FROM JOB a JOIN JOB b ON  
a.SUPERVISER_JOBNO = b.JOBNO  
JOIN CAREER c ON a.JOBNO = c.JOBNO  
JOIN EMP d ON c.EMPNO = d.EMPNO;
```

Рефлексивное соединение

Пример 25

Вывести код сотрудника, имя сотрудника и разницу в зарплате сотрудника и его руководителя

JOBNO	EMPNAME	SALARY_DIFF
1004	SMITH	2000
1003	SMITH	1000
1001	ALLEN	7500
1002	ALLEN	7000
1001	ALLEN	7500
1000	JONES	5500
1000	JOHN MARTIN	5500
1003	RICHARD MARTIN	1000
1004	RICHARD MARTIN	2000
1005	ALEX BOUSH	6200
1000	JOHN KLINTON	5500

Выражение CASE

- простое выражение CASE
- поисковое выражение CASE

Простое выражение CASE

```
CASE исходное_выражение  
WHEN выражение_сравнения  
THEN результирующее_выражение  
[WHEN выражение_сравнения  
THEN результирующее_выражение]  
[ELSE результирующее_выражение]  
END
```

Простое выражение CASE

Пример 33

```
SELECT DEPTNO,DEPTNAME,  
CASE DEPTNAME  
WHEN 'ACCOUNTING' THEN 'БУХГАЛТЕРИЯ'  
WHEN 'RESEARCH' THEN 'ОТДЕЛ ИССЛЕДОВАНИЙ'  
WHEN 'SALES' THEN 'ОТДЕЛ ПРОДАЖ'  
WHEN 'OPERATIONS' THEN 'ОПЕРАЦИОННЫЙ ОТДЕЛ'  
END AS OTDEL  
FROM DEPT;
```

Поисковое выражение CASE

CASE

WHEN условие THEN результирующее_выражение

[WHEN условие THEN результирующее_выражение]

[ELSE результирующее_выражение]

END

Поисковое выражение CASE

Пример 34

```
SELECT EMPNO, ENDDATE,  
CASE  
WHEN ENDDATE IS NOT NULL THEN 'уволен'  
ELSE 'работает'  
END AS STATE  
FROM CAREER;
```

Операторы DML

- INSERT – добавляет новые строки в таблицу БД
- UPDATE – обновляет существующие строки таблицы БД
- DELETE – удаляет строки из таблицы БД

Вставка строки в таблицу

```
INSERT INTO имя_таблицы [(список_полей)]  
VALUES (список_значений);
```

Пример 28

Добавить в таблицу EMP сотрудника

Вариант 1

```
INSERT INTO EMP VALUES ('1210', 'PETER WHITE', NULL);
```

Вариант 2

```
INSERT INTO EMP(EMPNO, EMPNAME) VALUES ('1210',  
'PETER WHITE');
```

Вставка строк в таблицу

```
INSERT INTO имя_таблицы [(список_полей)]  
SELECT запрос_выбора;
```

Пример 29

Зачислить всех уволенных сегодняшним числом работников на должность CLERK (код 1004) в отдел с номером 40 завтрашним числом

```
INSERT INTO CAREER  
SELECT 1004,EMPNO,40,SYSDATE+1,NULL  
FROM CAREER  
WHERE EMPNO NOT IN  
(SELECT EMPNO FROM CAREER  
WHERE JOBNO = 1004 OR ENDDATE < SYSDATE);
```

Вставка строки в таблицу

Пример 30

Добавить новую должность в таблицу JOB

```
INSERT INTO JOB
```

```
(JOBNO,JOBNAME,MINSALARY)
```

```
SELECT MAX(JOBNO)+1,'PROGRAMMER',3000
```

```
FROM JOB;
```

Модификация строк таблицы

UPDATE имя_таблицы

SET имя_поля = выражение [имя_поля = выражение...]

[WHERE условие];

Модификация строк таблицы

Таблица CAREER

JOBNO	EMPNO	DEPTNO	STARTDATE	ENDDATE
1004	7698	10	21-MAY-99	01-JUN-99
1003	7698	10	01-JUN-00	25-AUG-01
1006	7788		12-OCT-99	
1003	7369	20	21-MAY-05	01-JUN-06
1001	7499	30	02-JAN-03	31-DEC-05
1000	7654	20	21-JUL-99	01-JUN-04
1002	7499	30	01-JUN-06	25-OCT-08
1000	7566	20	12-OCT-06	
1004	7369	30	01-JUL-00	
1001	7499	10	01-JAN-08	
1005	7789	40	01-JAN-01	
1000	7790	40	01-OCT-01	

Пример 26

Уволить всех работающих сотрудников с сегодняшней даты кроме сотрудника занимающего должность PRESIDENT (код 1006) и сотрудников, которые занимают должность CLERK (код 1004)

Модификация строк таблицы

Пример 26

```
UPDATE CAREER
```

```
SET ENDDATE = SYSDATE
```

```
WHERE JOBNO NOT IN (1004,1006) AND ENDDATE IS NULL;
```

JOBNO	EMPNO	DEPTNO	STARTDATE	ENDDATE
1004	7698	10	21-MAY-99	01-JUN-99
1003	7698	10	01-JUN-00	25-AUG-01
1006	7788		12-OCT-99	
1003	7369	20	21-MAY-05	01-JUN-06
1001	7499	30	02-JAN-03	31-DEC-05
1000	7654	20	21-JUL-99	01-JUN-04
1002	7499	30	01-JUN-06	25-OCT-08
1000	7566	20	12-OCT-06	10-OCT-11
1004	7369	30	01-JUL-00	
1001	7499	10	01-JAN-08	10-OCT-11
1005	7789	40	01-JAN-01	10-OCT-11
1000	7790	40	01-OCT-01	10-OCT-11

Модификация строк таблицы

Пример 27

Установить для всех должностей, для которых в настоящее время нет работников, минимальный оклад, равный минимальной начисленной зарплате за январь 2008 года.

```
UPDATE JOB SET MINSALARY=
    (SELECT MIN(SALVALUE) FROM salary
     WHERE MONTH=1 AND YEAR=2008)
WHERE JOBNO NOT IN
    (SELECT DISTINCT JOBNO FROM CAREER
     WHERE ENDDATE IS NULL);
```

Удаление строк таблицы

```
DELETE FROM имя_таблицы  
[WHERE условие];
```

Пример 31

Удалить из таблицы CAREER все записи о сотрудниках уволившихся до 2000 года

```
DELETE FROM CAREER  
WHERE ENDDATE < TO_DATE('1-1-2000','dd-mm-yyyy')
```


Представления

Представление – это сохраненный запрос, который выдает пользователю БД требуемый набор данных из одной или нескольких таблиц

Функции представлений

Представления скрывают

- поля
- строки
- сложные операции с БД

Представления

CREATE [OR REPLACE] VIEW имя_представления AS
SQL-запрос;

Пример 32

Создать представление содержащее данные о сотрудниках работающих в отделе SALES

```
CREATE VIEW EMP_SALES AS
```

```
SELECT e.EMPNO, e.EMPNAME,s.MONTH,s.YEAR,  
s.SALEVALUE
```

```
FROM ((CAREER c JOIN DEPT d ON c.DEPTNO =  
d.DEPTNO) JOIN EMP e ON c.EMPNO = e.EMPNO) JOIN  
SALARY s ON c.EMPNO = s.EMPNO
```

```
WHERE d.DEPTNO = 'SALES' AND c.ENDDATE IS NOT  
NULL;
```