

# HIERARCHICAL QUERY and Cron Mask

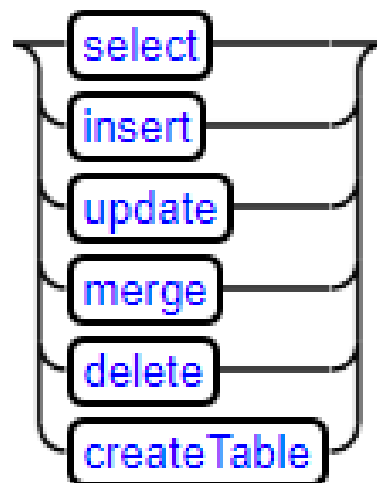
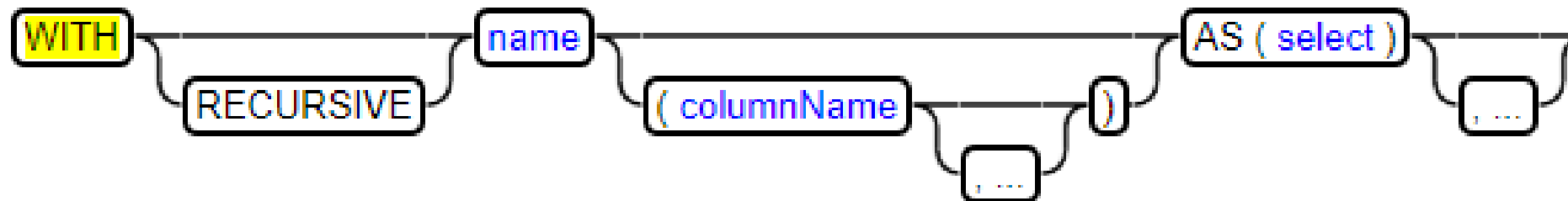
Regular meeting for get knowledge and tech in  
EDUC

## Рекурсивные (иерархические) запросы

- Рекурсивные запросы используются для обращения к иерархически связанным данным.
- В Oracle существует два синтаксиса:
  - 1. с использованием CONNECT BY
  - 2. с использованием WITH
- В H2 используется WITH

# Синтаксис WITH

**WITH**



1  
column

# Пример рекурсивного запроса в H2

- WITH RECURSIVE T(N) AS (
  - SELECT 1
  - UNION ALL
  - SELECT N+1 FROM T WHERE N<10
- )
- SELECT \* FROM T;

```
WITH RECURSIVE T(N) AS (  
    SELECT 1  
    UNION ALL  
    SELECT N+1 FROM T WHERE N<10  
)  
SELECT * FROM T;
```

N
1
2
3
4
5
6
7
8
9
10

(10 rows, 10 ms)

# Подготовка данных

- Создание таблицы
- `CREATE TABLE HQ_TEST(`
- `ID INT PRIMARY KEY,`
- `PARENT_ID INT,`
- `NAME VARCHAR(255)`
- `);`
- Вставка данных
- `INSERT INTO HQ_TEST values (1, null, '1 lvl');`
- `INSERT INTO HQ_TEST values (2, 1, '2 lvl');`
- `INSERT INTO HQ_TEST values (3, 1, '2 lvl');`
- `INSERT INTO HQ_TEST values (4, 3, '3 lvl');`
- `INSERT INTO HQ_TEST values (5, 3, '3 lvl');`
- `INSERT INTO HQ_TEST values (6, 4, '4 lvl');`
- `INSERT INTO HQ_TEST values (7, 4, '4 lvl');`
- `INSERT INTO HQ_TEST values (8, 5 , '4 lvl');`
- `INSERT INTO HQ_TEST values (9, null, '1 lvl');`
- `INSERT INTO HQ_TEST values (10, 9, '2 lvl');`

# Пример рекурсивного запроса в H2

- with recursive HQ(ID, PARENT\_ID, NAME) as
- (
  - select ID, PARENT\_ID, NAME from HQ\_TEST where ID = 9
  - union all
  - select t.ID, t.PARENT\_ID, t.NAME from HQ q
  - INNER join HQ\_TEST t where t.PARENT\_ID = q.ID
- )
- select \* from HQ;

```
with recursive HQ(ID, PARENT_ID, NAME) as
(
  select ID, PARENT_ID, NAME from HQ_TEST where ID = 9
  union all
  select t.ID, t.PARENT_ID, t.NAME from HQ q
  INNER join HQ_TEST t where t.PARENT_ID = q.ID
)
```

```
select * from HQ;
```

ID	PARENT_ID	NAME
9	<i>null</i>	1 lvl
10	9	2 lvl

(2 rows, 5 ms)

# Пример рекурсивного запроса в H2

- with recursive HQ(ID, PARENT\_ID, NAME) as
- (
  - select ID, PARENT\_ID, NAME from HQ\_TEST where ID = 7
  - union all
  - select t.ID, t.PARENT\_ID, t.NAME from HQ q
  - INNER join HQ\_TEST t where t.id = q.PARENT\_ID
- )
- select \* from HQ;

```
with recursive HQ(ID, PARENT_ID, NAME) as
(
    select ID, PARENT_ID, NAME from HQ_TEST where ID = 7
    union all
    select t.ID, t.PARENT_ID, t.NAME from HQ q
    INNER join HQ_TEST t where t.id = q.PARENT_ID
)
select * from HQ;
```

ID	PARENT_ID	NAME
7	4	4 lvl
4	3	3 lvl
3	1	2 lvl
1	<i>null</i>	1 lvl

(4 rows, 4 ms)

# CRON маска

- cron — классический демон (компьютерная программа в системах класса UNIX), использующийся для периодического выполнения заданий в определённое время.
- Маска cron состоит из 6 колонок, разделяемых пробелами или табуляторами. Первые пять колонок задают время выполнения (Минута, Час, День, Месяц, День недели), в них может находиться число, список чисел, разделённых запятыми, диапазон чисел, разделённых дефисом, символы '\*' или '/'.
  - Например:
  - 30 15 \* \* \* \*
  - 15 08,17 \* \* 1-5 \*
  - 0 19 4 \* \* \*
  - 0 20 \*/5 \* \* \*
  - 0 20 \* \* 6 \*



# Расшифровка Cron маски

- 30 15 \* \* \* \* - ежедневно в 15:30
- 15 08,17 \* \* 1-5 \* - с понедельника по пятницу в 8:15 и 17:15
- 0 19 4 \* \* \* - в 19:00 4 числа каждого месяца
- 0 20 \*/5 \* \* \* - в 20:00 каждые 5 дней
- 0 20 \* \* 6 \* - в 20:00 каждую субботу
- | | | | | |
- | | | | | +--- Годы (диапазон: 1900-3000)
- | | | | +----- Дни недели (диапазон: 1-7)
- | | | +----- Месяцы (диапазон: 1-12)
- | | +----- Дни месяца (диапазон: 1-31)
- | +----- Часы (диапазон: 0-23)
- +----- Минуты (диапазон: 0-59)

# Расшифровка Cron маски

Значение	Диапазон	Дополнительно
минуты	0-59	
часы	0-23	
дни месяца	1-31	
месяцы	1-12	можно задавать и в 3-х буквенном варианте
дни недели	0-6	можно задавать и в 3-х буквенном варианте (0=воскресенье)

# Пример Cron

- `INSERT INTO CRON_EXAMPLE(ID, CATEGORY_ID, SUMM, START_DATE, END_DATE, CRON_MASK, REPEAT_NUMBER)`
- `values (1, '1.2', 100, FORMATDATETIME(PARSEDATETIME('01.01.2018', 'dd.MM.yyyy'), 'yyyy-MM-dd'), null, '30 15 * * * *', null);`
- `INSERT INTO CRON_EXAMPLE(ID, CATEGORY_ID, SUMM, START_DATE, END_DATE, CRON_MASK, REPEAT_NUMBER)`
- `values (2, '2.1', 5, DATE '2018-01-09', DATE '2019-05-31', '15 08,17 * * 1-5 *', null);`
- Преобразование строки в дату в H2
- `FORMATDATETIME(PARSEDATETIME('01.01.2018', 'dd.MM.yyyy'), 'yyyy-MM-dd')`
- `DATE '2018-01-09'`