

ОСНОВНЫЕ ОБЪЕКТЫ БАЗЫ ORACLE

Графеева Н.Г.

2020

Основные объекты ORACLE

- Таблицы
- Правила целостности
- Индексы
- Представления
- Секвенции
- Процедуры
- Функции
- Пакеты
- Триггеры
- Задания
- Синонимы
- Соединения
- И многое другое....

Таблицы

При определении таблицы главное – определение имен столбцов таблицы и задание их типов.

Основные типы данных, используемые при создании таблиц:

CHAR(размер)

Символьные данные фиксированной длины. Максимальный размер - 255 байт. Умалчиваемый размер - 1 байт.

VARCHAR2 (размер)

Символьные данные переменной длины. Максимальный размер - 2000 байт. Умалчиваемый размер - 1 байт.

NUMBER(p,s)

Числовые данные переменной длины. Точность p (общее количество цифр) может задаваться от 1 до 38. Масштаб s (число цифр после десятичной точки) может быть от -84 до 127

DATE

Значение даты и времени (с точностью до секунд) в интервале от 1 января 4712 г. до н.э. до 31 декабря 4712 г. н.э.

TIMESTAMP

Дата и время с точностью до миллисекунд

=====

Над всеми типами данных предусмотрен достаточный набор операций, но не все операции над типами данных так очевидны. Наиболее часто возникают проблемы с типом DATE.

Пример (основные операции с датами)

```
SELECT SYSDATE, CURRENT_DATE,  
       SYSDATE + 1, CURRENT_DATE_PLUS_1_DAY,  
       SYSDATE + 1/24, CURRENT_DATE_PLUS_1_HOUR,  
       EXTRACT(YEAR FROM SYSDATE), CURRENT_YEAR,  
       EXTRACT(MONTH FROM SYSDATE), CURRENT_MONTH,  
       TO_CHAR(SYSDATE, 'DD/MM/YYYY HH24:MI:SS') ,  
       TO_DATE('01/01/2015 12:20:25', 'DD/MM/YYYY HH24:MI')  
FROM DUAL
```

Работа с таблицами (SQL DDL)

```
CREATE TABLE table-name (field-name type [ options ]  
[, field-name type [ options ] ]*)
```

```
=====
```

```
ALTER TABLE table-name {ADD|DROP|ALTER}[COLUMN]  
(field-name [type] [ options ])
```

```
=====
```

```
DROP TABLE table_name [CASCADE CONSTRAINTS]
```

Примечание : опции – это правила целостности, значения по умолчанию и пр.

Системные представления (словарь данных) для просмотра таблиц

- USER_TABLES
- ALL_TABLES
- DBA_TABLES (только для администраторов)

Что еще полезно знать про таблицы?

- При задании таблиц можно задавать параметры их физического хранения: PCTFree и PCTUsed
- Эти параметры позволяют оптимально распределять дисковое пространство для задач редактирования или чтения таблиц.

Задание 1

1.Взгляните на содержимое системных табличных представлений, доступных в вашей схеме (в APEX, SQL Commans).

Примечание: В системном представлении ALL_TABLES выберите не более 10 первых строк (where rownum <= 10).

2.Определите количество доступных вам таблиц (через системное представление ALL_TABLES).

3.Выведите содержимое какой-нибудь таблицы, которая видна в ALL_TABLES, но не присутствует в USER_TABLES (не более 10 записей).

Правила целостности

В Oracle используются следующие правила целостности:

NOT NULL	- запрет пустых значений
UNIQUE	- контроль уникальности
PRIMARY KEY	- первичный ключ
FOREIGN KEY	- внешний ключ
CHECK	- контроль допустимых значений

Работа с правилами целостности (SQL DDL)

```
ALTER TABLE table_name {ADD|DROP|MODIFY} CONSTRAINT [const_name]  
[const_definition]
```

```
ALTER TABLE table_name RENAME CONSTRAINT const_name TO const_name
```

```
ALTER TABLE table_name {ENABLE|DISABLE} [CONSTRAINT] [const_name]  
[const_definition]
```

Примеры

```
ALTER TABLE emp ADD CONSTRAINT emp_pk PRIMARY KEY (empno);
```

```
=====
```

```
ALTER TABLE EMPLOYEES ADD CONSTRAINT EMP_JOB_FK FOREIGN KEY (JOB_ID)  
REFERENCES JOBS (JOB_ID)
```

```
=====
```

```
ALTER TABLE emp DROP CONSTRAINT dept_fkey
```

```
=====
```

```
ALTER TABLE dept DROP UNIQUE (dname, loc)
```

```
=====
```

```
ALTER TABLE dept RENAME CONSTRAINT dname_ukey TO dname_unikey
```

```
=====
```

```
ALTER TABLE dept DISABLE CONSTRAINT dname_ukey
```

```
=====
```

```
ALTER TABLE dept ENABLE dname_ukey
```

Рекомендация

- При работе с правилами целостности настоятельно рекомендуется именовать эти правила своими руками и не рассчитывать на автоматическое именование со стороны ORACLE. Автоматическое именование правил целостности может создать проблемы при работе с несколькими версиями базы (тестовой, рабочей, базой для разработки и т.п.)

Системные представления для просмотра правил целостности

- USER_CONSTRAINTS
- ALL_CONSTRAINTS
- DBA_CONSTRAINTS (только для администраторов)

Задание 2

- Взгляните на содержимое системных представлений правил целостности в вашей схеме (в APEX, SQL Commands). Сколько правил целостности в вашей схеме?
- Добавьте правило целостности (без явного именованя), ограничивающее размер зарплаты: $(500 \leq \text{sal} \leq 5000)$. Убедитесь, что оно будет отображаться в соответствующем системном представлении. Напишите запрос, который выводит имя этого правила целостности.
- Удалите это правило целостности.
- Добавьте это же правило целостности с явным именованием. Убедитесь, что оно будет отображаться в соответствующем системном представлении.

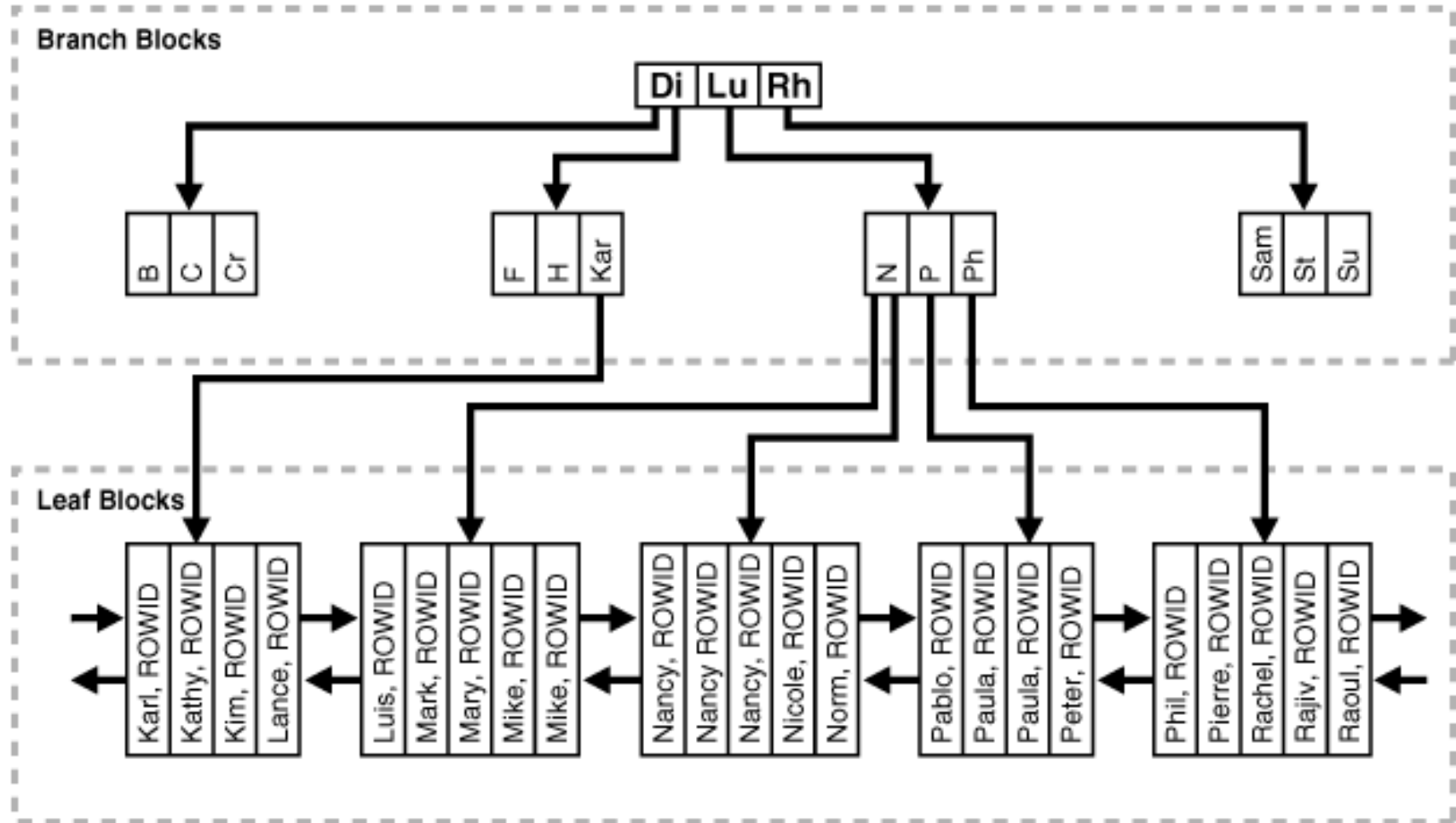
Индексы

Индекс – это структура, связанная с таблицей и используемая в первую очередь для ускорения доступа к строкам таблицы.

Основные формы организации индексов в ORACLE:

- B-tree индексы
- BITMAP индексы

B-tree индекс



BITMAP индекс (содержимое исходной таблицы)

CUSTOMER #	MARITAL_ STATUS	REGION	GENDER
101	single	east	male
102	married	central	female
103	married	west	female
104	divorced	west	male
105	single	central	female
106	married	central	female

BITMAP индекс (структура индекса)

REGION='east'	REGION='central'	REGION='west'
1	0	0
0	1	0
0	0	1
0	0	1
0	1	0
0	1	0

ВІТМАР индекс (обработка запроса)

```
select count(*) from FROM CUSTOMER WHERE MARITAL_STATUS = 'married' AND REGION IN ('central','west')
```

status = 'married'	region = 'central'	region = 'west'
0	0	0
1	1	0
1	0	1
0	0	1
0	1	0
1	1	0

Создание и удаление индексов (SQL DDL)

```
CREATE [UNIQUE] [BITMAP] INDEX index-name  
ON table-name(field-name {,field-name}*)
```

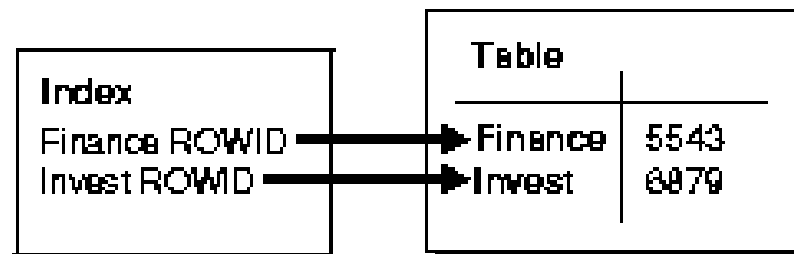
```
DROP INDEX index-name
```

Какие плюсы и минусы сопровождает создание индексов?

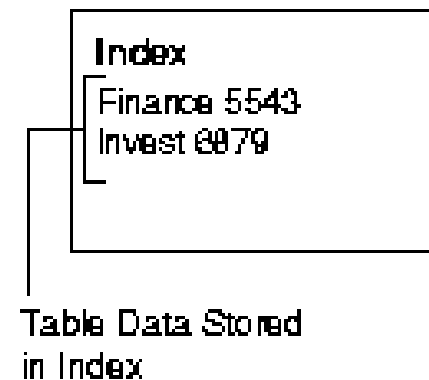
- Индексы замедляют добавление новых записей.
- Индексы увеличивают объемы базы на диске.
- + Индексы ускоряют выборку данных.

Таблицы, организованные как индексы

Regular Table and Index



Index-Organized Table



Создание таблиц-индексов (SQL DDL)

CREATE TABLE table-name

(field-name type [options] [UNIQUE (field-list)]

{,field-name type[options] [UNIQUE (field-list)]}*)

ORGANIZATION INDEX

Пример

```
CREATE TABLE COUNTRIES  
(  
  COUNTRY_ID  CHAR(2) CONSTRAINT COUNTRY_ID_NN NOT NULL,  
  COUNTRY_NAME VARCHAR2(40),  
  REGION_ID   NUMBER,  
  CONSTRAINT COUNTRY_C_ID_PK PRIMARY KEY (COUNTRY_ID)  
)  
ORGANIZATION INDEX
```


Системные представления для просмотра индексов

- USER_INDEXES
- ALL_INDEXES
- DBA_INDEXES (доступно администратору)

Задание 3

- Взгляните на содержимое представлений, связанных с индексами в своей схеме (в APEX, SQL Commands).
- Подсчитайте количество индексов в своей схеме (напишите запрос).
- Создайте индексно-организованную таблицу DEPT1 (аналогичную таблице DEPT). Заполните ее содержимым из DEPT.

Зачетное задание 2(8 баллов)

- Оформите все задания в виде одного ***переиспользуемого скрипта***.

Результат предъявите или отправьте по адресу N.Grafeeva@spbu.ru. Тема письма – DB_Application_2020_job2.

Примечание: задание должно быть сдано (отправлено) в течение 2 недель. За более позднее отправленное будут сниматься штрафные баллы (по баллу за каждые 2 недели).