

1. Типы данных Oracle
2. Целостность данных
3. Словарь данных

Правила Э.Ф.Кодда

1. Данные хранятся в столбцах и строках таблиц
2. Доступ к данным можно получить, указав имя таблицы, имя столбца и ключ
3. СУБД должна обрабатывать пропущенные значения (пустые данные). Для обозначения пустых данных используется ключевое слово NULL
4. СУБД должна включать оперативный каталог, содержащий сведения о самой базе данных
5. Для определения данных, их обработки и других операций определяется специальный подязык. В настоящее время таким языком является SQL
6. Должны поддерживаться представления таблиц, или виртуальные таблицы, которые строятся динамически по запросам
7. Должна быть включена поддержка транзакций. Транзакция обеспечивает коллективное изменение или отмену всех связанных изменений данных. Транзакции имеют четыре свойства: атомарность, согласованность, изолированность и продолжительность (ACID). Атомарность – транзакция либо выполняется полностью, либо не выполняет ничего. Согласованность – с каждой транзакцией БД переходит из одного согласованного состояния в другое согласованное состояние данных. Изолированность – изменения, происходящие в процессе транзакции, не видны пользователю до завершения транзакции. Продолжительность – сделанные в процессе транзакции изменения должны сохраниться в БД
8. Физическое хранение данных отделено от пользователя. Пользователь имеет дело только с логической структурой БД
9. Логическую структуру данных можно изменять с минимальным воздействием на пользователей и программы
10. Правила целостности данных хранятся в каталоге БД. Любые их изменения не должны влиять на прикладные программы
11. Приложения должны работать в распределенной среде (когда данные хранятся в различных местах)
12. СУБД должна обеспечивать безопасность и целостность базы данных

СТРУКТУРА БАЗЫ ДАННЫХ

1. Типы данных Oracle

Набор символов

- Однобайтовые фиксированной длины (ASCII)
- Многобайтовые фиксированной длины (Unicode UTF-16, каждый символ представлен двумя байтами)
- Многобайтовые переменной длины (Unicode UTF-8, количество байтов на один символ не всегда одинаково – от одного до трех)
- *Набор символов базы данных.* Используется для представления значений столбцов типа CHAR и VARCHAR2, имен таблиц, столбцов, переменных PL/SQL, SQL-инструкций, большинства строковых литералов и других подобных идентификаторов.
- *Набор символов национального языка.* Используется для представления значений столбцов типа NCHAR и NVARCHAR2 и строковых литералов с префиксом N.

```
SELECT * FROM nls_database_parameters  
WHERE parameter IN ('NLS_CHARACTERSET','NLS_NCHAR_CHARACTERSET');
```

1.1. Символьные типы данных

1.1.1. Типы данных CHAR

В столбцах типа CHAR хранятся строки фиксированной длины.

Строка CHAR имеет длину (в байтах) между 1 и 255 (в версиях до Oracle8i) или 2000 (начиная с Oracle8i).

имя_столбца CHAR (длина)

1.1.2. Типы данных VARCHAR2

Столбцы типа VARCHAR2 хранят строки переменной длины.

При создании таблицы со столбцом VARCHAR2 задается максимальная длина столбца (в байтах, а не в символах) от 1 до 2000 байтов (в версиях до Oracle8i) или 4000 (начиная с Oracle8i).

имя_столбца VARCHAR2 (длина)

1.1.3. Тип данных VARCHAR

VARCHAR является просто синонимом для **VARCHAR2**.

1.1.4. Столбцы символьных типов и символьные наборы NLS

Поддержка национального языка (NLS) позволяет использовать различные символы набора для хранения данных в полях типа **NCHAR** (с версии 9.0) и **NVARCHAR2**.

В версиях до Oracle9i в качестве национального набора символов мог использоваться любой однобайтовый или многобайтовый набор. Начиная с Oracle9i, в полях типа **NCHAR** и **NVARCHAR2** используются только UTF-8 или UTF-16.

1.1.5. Пустые строки

СУБД Oracle интерпретирует пустые строки как значения **NULL**, что не согласуется со стандартом ANSI SQL. Это относится к строкам типа **VARCHAR2**. Поля типа **CHAR** не могут содержать пустые строки по определению, т.к. это поля фиксированной длины.

1.1.6. Дополнительные типы

STRING, **CHARACTER VARYING**, **NATIONAL CHARACTER VARYING** и другие

1.2. Числовые типы данных

1.2.1. Тип данных NUMBER

NUMBER хранит числа с фиксированной и плавающей точками. Максимальная точность – до 38 знаков.

- положительные числа от 1×10^{-130} до $9.99...9 \times 10^{125}$ (до 38 значащих цифр);
- отрицательные числа от -1×10^{-130} до $-9.99...9 \times 10^{125}$ (до 38 значащих цифр);

- ноль;
- положительная и отрицательная бесконечности (их существование необходимо только для импорта данных из базы данных версии 5).

имя_столбца NUMBER [(точность, масштаб)]

| Входные данные | Спецификация | Хранение |
|----------------|--------------|-------------------------------|
| 7,456,123.89 | NUMBER | 7 456 123.89 |
| 7,456,123.89 | NUMBER(*,1) | 7 456 123.9 |
| 7,456,123.89 | NUMBER(9) | 7 456 124 |
| 7,456,123.89 | NUMBER(9,2) | 7 456 123.89 |
| 7,456,123.89 | NUMBER(9,1) | 7 456 123.9 |
| 7,456,123.89 | NUMBER(6) | (Неверно, превышена точность) |
| 7,456,123.89 | NUMBER(7,-2) | 7 456 100 |

1.2.2. Дополнительные типы

FLOAT, REAL, NUMERIC, DECIMAL, INTEGER и другие совместимые с ANSI

С версии 10.1 – BINARY_FLOAT, BINARY_DOUBLE

1.3. Типы даты-времени

1.3.1. Тип DATE

Диапазон дат от 1 января 4712 года до н.э. до 31 декабря 4712 года н.э.

имя_столбца DATE

Внутренний формат для хранения значений дат – поля фиксированной длины в 7 байт: век, год, месяц, день, час, минута и секунда.

TO_DATE ('November 13, 1992', 'MONTH DD, YYYY')

```
INSERT INTO birthdays (bname, bday) VALUES  
(‘ANDY’, TO_DATE(‘09.10.04 11:56 AM’, ‘DD.MM.YY HH:MI AM’));
```

Время: в формате AM – до полудня (до 12 часов дня)
в формате PM – после полудня (после 12 часов дня)

Текущая дата – 03-AUG-2000 14:35:21

```
select trunc(sysdate) from dual;      select round(sysdate) from dual;
```

```
TRUNC(SYS  
-----  
03-AUG-00
```

```
ROUND(SYS  
-----  
04-AUG-00
```

1.3.2. Типы TIMESTAMP

С версии 9.2 – TIMESTAMP, TIMESTAMP (n), TIMESTAMP WITH TIME ZONE, TIMESTAMP WITH LOCAL TIME ZONE

1.3.3. Интервалы времени

С версии 9.2 – INTERVAL YEAR TO MONTH, INTERVAL DAY TO SECOND

1.4. Тип данных LONG

Значения типа LONG – это строка переменной длины (до 2 Гбайт).

1.5. Тип данных RAW и LONG RAW

RAW – для хранения двоичных данных объемом до 255 байтов.

LONG RAW – эквивалент поля LONG для двоичных данных (до 2 Гбайт).

1.6. Тип данных ROWID

Данные типа ROWID представляют собой строку шестнадцатиричных цифр (до версии Oracle8) или строку символов (с версии Oracle8), которая обозначает адрес строки таблицы в базе данных.

Строка типа ROWID для Oracle7 имеет следующий формат:

block.row.file,

где **block** – номер блока данных в файле данных; **row** – номер строки, **file** – номер файла.

Начиная с Oracle8 формат ROWID принял вид

object file bloc slot,

где **object** – номер объекта, **file** – номер файла в табличном пространстве, **bloc** – номер блока в файле, **slot** – номер слота (строки) в блоке.

ROWID (Oracle7): 0000000F.0000.0002.

ROWID (Oracle8): AAAN2pAAIAAAAoSAAA

DECLARE

sal_rowid ROWID;

sal_comm NUMBER;

BEGIN

-- Выборка данных, которые мы хотим модифицировать

SELECT rowid, comm INTO sal_rowid, sal_comm

FROM sal

WHERE sname='Peel' AND city='London';

/* Вычисление новых комиссионных */

UPDATE sal

SET comm = sal_comm + 0.01

WHERE rowid = sal_rowid;

END;

1.7. Тип данных MLSLABEL

Этот тип данных используется с опцией **Trusted Oracle** для внедрения метки уровня доступа.

1.8. Большие объекты – LOB

**В версии Oracle8 появились большие объекты – LOB (Large Objects).
Позволяют хранить и обрабатывать данные больших объемов (до 128 Тбайт).**

- **CLOB – символьные объекты**
 - **BLOB – двоичные объекты**
 - **BFILE – указатели файлов LOB, внешних по отношению к БД**
-
- 1) В таблице может быть несколько столбцов CLOB, BLOB и BFILE**
 - 2) В столбцах таблицы хранятся только небольшие указатели на объекты LOB**
 - 3) Данные основной таблицы и объекты LOB могут храниться в разных местах (дисках) базы данных**
 - 4) Приложения могут обращаться к частям объектов LOB**

2. Целостность данных

2.1. Правила целостности.

Правила целостности – это декларации о данных таблицы, которые всегда выполняются:

| | |
|--------------------|---|
| NOT NULL | Не позволяет в столбцах хранить неопределенные (пустые или отсутствующие) значения. |
| UNIQUE | Не позволяет значениям столбца или набора столбцов одной таблицы дублироваться. |
| PRIMARY KEY | Не позволяет значениям столбца или набора столбцов дублироваться и быть неопределенными. |
| FOREIGN KEY | Требует, чтобы каждое значение столбца или набора столбцов в одной таблице, соответствовало значению столбца или набора столбцов, объявленных в другой с правилом целостности UNIQUE или PRIMARY KEY (правила целостности FOREIGN KEY также задают действия ссылочной целостности, которые нужно выполнять над зависимыми данными при изменении данных, на которые они ссылаются). |
| CHECK | Отменяет значения, которые не удовлетворяют логическому выражению, заданному правилом. |

2.1.1. Ключи

Ключ – это столбец или набор столбцов

| | |
|------------------------|---|
| Первичный ключ | Столбец или набор столбцов, заданных в определении правила целостности PRIMARY KEY . Значения первичного ключа уникально идентифицируют строки таблицы. Для таблицы может быть определен только один первичный ключ. |
| Уникальный ключ | Столбец или набор столбцов, включенных в определение правил целостности UNIQUE . |
| Внешний ключ | Столбец или набор столбцов, включенных в определение ссылочного правила целостности. |
| Ссылочный ключ | Уникальный или первичный ключ той же самой или другой таблицы, на который ссылается внешний ключ. |

2.2. Триггеры базы данных

Триггер базы данных – это хранимая процедура, которая выполняется неявно (срабатывает), когда выдаются, например, команды INSERT, UPDATE или DELETE для связанных с триггером таблиц.

Триггер базы данных определяется для того, чтобы обеспечить правила целостности, не применяя их к данным, которые уже существуют в таблице.

3. Словарь данных

Словарь данных базы данных Oracle – это набор таблиц и представлений, которые используются пользователями только для просмотра информации о базе данных.

Словарь данных создается тогда, когда создается база данных.

3.1. Системные таблицы и представления

Системные таблицы и представления, составляющие словарь данных, всегда принадлежат пользователю Oracle по имени SYS, и права на некоторые из системных таблиц (представлений) имеет каждый пользователь.

```
SELECT * FROM DICT; – список представлений, доступных
                        пользователю
```

3.1.1. Представления USER_

Информация об объектах, принадлежащих пользователю.

```
SELECT * FROM DICT
      WHERE TABLE_NAME LIKE 'USER_%'
      ORDER BY TABLE_NAME;

SELECT * FROM user_tables;
```

3.1.2. Представления ALL_

Информация об объектах, к которым пользователь имеет доступ (т.е. к своим и чужим).

```
SELECT * FROM all_constraints;
```

3.1.3. Представления DBA_

Информация об объектах, к которым имеет доступ администратор базы данных.

```
SELECT * FROM dba_cons_columns;
```

3.1.4. Представления V\$

Служебная информация о состоянии базы данных.

3.1.5. Другие представления

Представления, отвечающие требованиям международных стандартов.

```
SELECT * FROM tabs;
```

4. Online-документация

\\Oracle\Ora_Doc\index.htm

Oracle9i Database Online Documentation
(Release 2 (9.2))

[Help](#) [Feedback](#)

Go directly to the most popular information:

- Learn about [getting started](#) with the Oracle9i Database
- Browse the [list of books](#), including PDF for printing.
- [SQL and PL/SQL](#) syntax and examples.
- Look up an [error message](#).
- [Initialization parameters](#).
- [Catalog views / Data dictionary views](#).
- Find a short definition in the [Master Glossary](#).

Look up a term in the Master Index:

| | | | | | | | |
|-------------------------|--------------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| Symbols | Numerals | A | B | | | | |
| C | D | E | F | G | H | I | J |
| K | L | M | N | O | P | Q | R |
| S | T | U | V | W | X | Y | Z |