



Физическая организация данных в БД

Основные концепции.



Механизмы среды хранения и архитектура СУБД

При добавлении записи:

- ☞ поиск свободного места для размещения новой записи в пространстве памяти;
- ☞ выделение необходимого объема памяти под запись;
- ☞ размещение записи в отведенном месте памяти;
- ☞ формирование связей с другими записями (зависит от модели данных).

При поиске записи:

- ☞ поиск места размещения записи в пространстве памяти по заданным значениям атрибутов;
- ☞ выборка записи для обработки в оперативную память (в буфер данных).

При изменении атрибутов записи:

- ☞ поиск записи и считывание её в ОП;
- ☞ изменение значений атрибута (атрибутов) записи;
- ☞ сохранение записи на диск.

При удалении записи:

- ☞ удаление записи с освобождением памяти (*физическое удаление*) или без освобождения (*логическое удаление*);
- ☞ разрушение связей с другими записями (механизм зависит от модели данных).

Структура хранимых данных

Единица хранения данных в БД – **хранимая запись**.

Хранимая запись состоит из двух частей:

1. *Служебная часть*. Используется для идентификации записи, задания её типа, хранения признака логического удаления, для кодирования значений элементов записи, для установления структурных ассоциаций между записями и проч. Никакие пользовательские программы не имеют доступа к служебной части хранимой записи.
2. *Информационная часть*. Содержит значения элементов данных. Поля хранимой записи могут иметь *фиксированную* или *переменную длину*. Хранение полей переменной длины осуществляется одним из двух способов: размещение полей через разделитель или хранение размера значения поля. Наличие полей переменной длины позволяет не хранить незначащие символы и снижает затраты памяти на хранение данных; но при этом увеличивается время на извлечение записи.

Структура хранимых данных

Каждой хранимой записи БД система присваивает внутренний идентификатор, называемый (по стандарту CODASYL) **ключом базы данных** (КБД).

В Oracle используется термин *идентификатор строки*, RowID.

Значение КБД формируется системой при размещении записи и содержит информацию, позволяющую однозначно определить место размещения записи (преобразовать значение КБД в адрес записи).

Примеры:

- 1) Формат DBF: 1 таблица – 1 файл, записи фиксированной длины – в качестве КБД выступает последовательный номер записи в файле (относительная адресация).
- 2) СУБД Oracle – совокупность номера экстента, блока и номера строки в блоке (относительная адресация).
- 3) Абсолютный адрес в памяти (СУБД Adabas).

Управление пространством памяти

Для обеспечения естественной структуризации хранимых данных, более эффективного управления ресурсами и/или для технологического удобства всё пространство памяти БД обычно разделяется на части (области, сегменты и др.). В каждой *области памяти*, как правило, хранятся данные одного объекта БД (одной таблицы). Сведения о месте расположения данных таблицы (ссылка на область хранения) СУБД хранит в словаре-справочнике данных (ССД). Области разбиваются на пронумерованные *страницы (блоки)* фиксированного размера. В большинстве систем обработку данных на уровне страниц ведёт операционная система (ОС), а обработку записей внутри страницы обеспечивает только СУБД.

Страницы представляются в среде ОС блоками внешней памяти или секторами, доступ к которым осуществляется за одно обращение. Некоторые СУБД позволяют управлять размером страницы (блока) для базы данных.

Структура
страницы
памяти

(заголовок)

Управление пространством памяти

Способы управления свободным пространством памяти на страницах:

- ведение списков свободных участков;
- динамическая реорганизация страниц.

При **динамической реорганизации страниц** записи БД плотно размещаются вслед за заголовком страницы, а после них расположен свободный участок. Смещение начала свободного участка хранится в заголовке страницы.

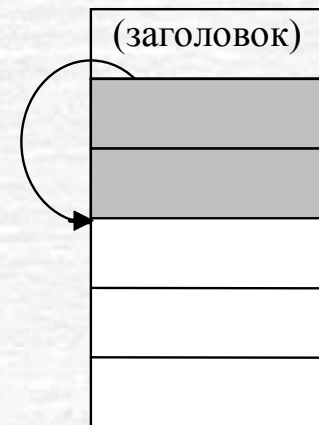
Достоинство такого подхода – отсутствие фрагментации.

Недостатки:

- ✓ Адрес записи может быть определён с точностью до адреса страницы, т.к. внутри страницы запись может перемещаться.
- ✓ Поиск места размещения новой записи может занять много времени. Система будет читать страницы одну за другой до тех пор, пока не найдёт страницу, на которой достаточно места для размещения новой записи.

Ведение инвентарных страниц.

динамическая
реорганизация
страниц



Управление пространством памяти

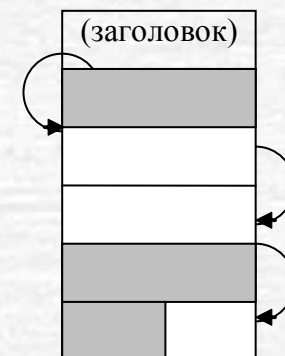
Ведение **списков свободных участков**.

Здесь можно рассмотреть два варианта:

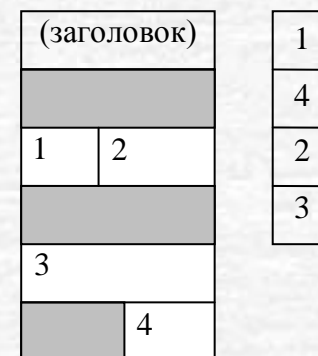
1. Ссылка на первый свободный участок на странице хранится в заголовке страницы, и каждый свободный участок хранит ссылку на следующий (или признак конца списка). Каждый освобождаемый участок включается в список свободных участков на странице.
2. Списки свободных участков реализуются в виде отдельных структур. Эти структуры также хранятся на отдельных *инвентарных страницах*. Каждая инвентарная страница относится к области (или группе страниц) памяти и содержит информацию о свободных участках в этой области. Список ведётся как стек, очередь или упорядоченный список. В последнем случае упорядочение осуществляется по размеру свободного участка, что позволяет при размещении новой записи выбирать для неё наиболее подходящий по размеру участок.

Достоинства и недостатки.

списки свободных участков на странице



списки свободных участков в виде отдельных структур

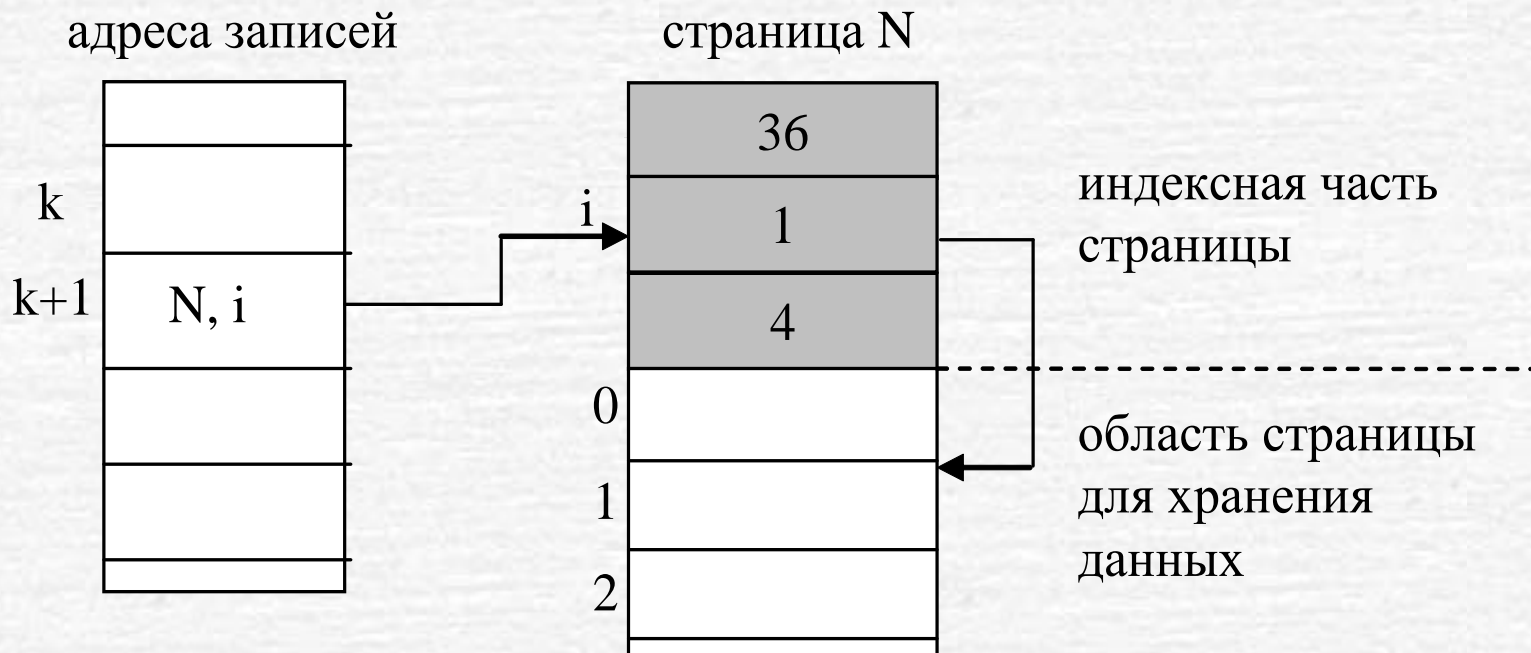


Виды адресации хранимых записей

Рассмотрим три вида адресации: прямую, косвенную и относительную.

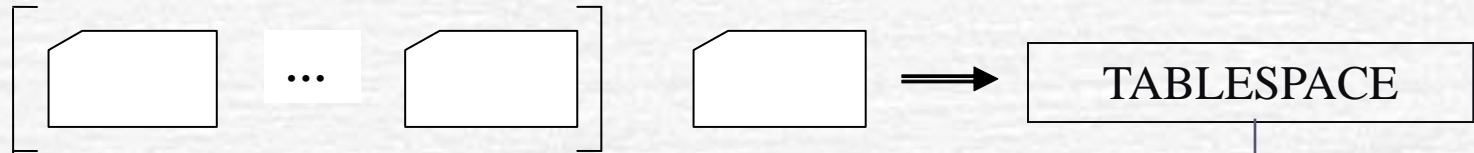
- **Прямая адресация** предусматривает указание непосредственного местоположения записи в пространстве памяти (например, в системе ADABAS). Недостатки: большой размер адреса; прямая адресация не позволяет перемещать записи в памяти без изменения КБД, что ведёт к фрагментации памяти.
- **Косвенная адресация.** В качестве КБД выступает не сам "адрес записи", а адрес места хранения "адреса записи".
- **Относительная адресация.** Общий принцип относительной адресации заключается в том, что адрес отсчитывается от начала той области памяти, которую занимают данные объекта БД. Если память разбита на страницы (блоки), то адресом может выступать номер страницы (блока) и номер записи на странице (или смещение от начала страницы). В случае относительной адресации перемещение записи приведёт к изменению КБД и необходимости корректировки индексов, если они есть.

Пример косвенной адресации



Часть адресного пространства страницы выделяется под индекс страницы. Число статей (слотов) в нём одинаково для всех страниц. В качестве КБД запись выступает совокупность номера нужной страницы и номера требуемого слота в индексе этой страницы (значения N, i). В i -м слоте на N -й странице хранится собственно адрес записи (смещение от начала страницы).

Основные физические структуры Oracle



Файлы – это файлы операционной системы, выделенные для хранения базы данных.

Табличная область (TABLESPACE, табличное пространство) – область памяти, предназначенная для хранения всех объектов БД. Табличная область имеет имя и занимает один или более файлов операционной системы. Создается командой `CREATE TABLESPACE`.

Сегмент (SEGMENT) – область памяти, занимаемая данными одного объекта БД. Имя совпадает с именем объекта.

Экстент (EXTENT) – непрерывная область памяти, относящаяся к одному сегменту.

Блок (BLOCK) – область памяти, которая считывается и записывается на диск за одно физическое чтение.

`db_block_size` –
размер блока

Формат блока данных Oracle

Заголовок (общий и переменный)

- Заголовок содержит общую информацию блока, такую как адрес блока и тип сегмента (сегмент данных, сегмент индекса или сегмент отката). Заголовок составляет накладные расходы блока, которые имеют переменный размер. В среднем, суммарные накладные расходы фиксированной и переменной частей блока составляют от 84 до 107 байт.

Оглавление таблиц

- Часть блока, составляющая оглавление таблиц, содержит информацию о том, какие таблицы имеют строки в этом блоке.

Оглавление строк

- Эта часть блока содержит информацию о действительных строках в блоке (включая адреса каждой порции строки в области данных строк). После того, как в оглавлении строк распределено пространство, это пространство не освобождается при удалении строки. Это пространство используется повторно лишь тогда, когда в блок вставляются новые строки.

Данные строк

- Эта порция блока содержит данные таблицы или индекса.

Свободное пространство

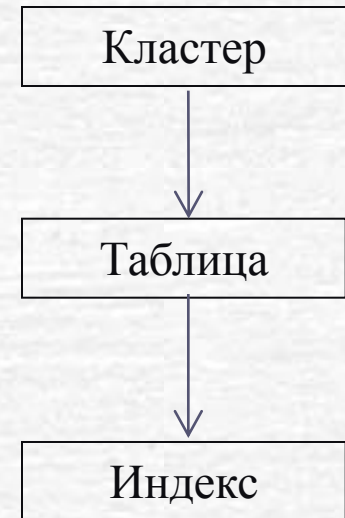
- Свободное пространство в блоке используется для вставки новых строк и для обновлений строк, требующих дополнительного пространства.

Структура блока

заголовок
оглавление таблиц
оглавление строк
свободное пространство
данные строк

Основные объекты Oracle

- ❖ **Кластер (CLUSTER)** – объект, задающий способ совместного хранения данных нескольких таблиц, содержащих информацию, обычно обрабатываемую совместно. Создается командой CREATE CLUSTER. Включает таблицы с данными.
- ❖ **Таблица (TABLE)** является базовой структурой реляционной модели. Таблица может быть пустой или состоять из одной или более строк значений атрибутов.. Создается командой CREATE TABLE, может быть создана в кластере.
- ❖ **Индекс (INDEX)** – это объект базы данных, создаваемый для повышения производительности выборки данных. Индекс создается для столбца (столбцов) таблицы и обеспечивает более быстрый доступ к данным этой таблицы за счет упорядочения данных столбца (столбцов) по значению. Создается командой CREATE INDEX.



Кластеры, таблицы и индексы называются *объектами, занимающими память*, т.к. в них хранятся фактографические данные.

Создание таблицы в Oracle

```
CREATE TABLE [имя_схемы.]имя_таблицы  
( имя_столбца тип_данных_столбца [ DEFAULT выражение] [CONSTRAINT  
имя_ограничения_целостности] [ограничение_целостности_столбца ...] } ,...)  
[ ORGANIZATION INDEX ]  
[ { CLUSTER имя_кластера ( имя_столбца [, ...]) |  
{PCTFREE целое | PCTUSED целое | INITRANS целое | MAXTRANS целое |  
TABLESPACE имя_табличной_области | STORAGE размер_памяти |  
{ RECOVERABLE | UNRECOVERABLE } } ... ]  
[PARALLEL возможность_параллельной_обработки ]  
[ {ENABLE проверяемые_ограничения_целостности |  
DISABLE игнорируемые_ограничения_целостности }... ]  
[AS запрос] [CACHE | NOCACHE];
```

UNRECOVERABLE (без заполнения журнала) не может быть указано без AS SELECT.

Нельзя указать UNRECOVERABLE для кластера или кластеризованной таблицы.

Параметры распределения памяти в Oracle

Размер памяти (для всех объектов, занимающих память) устанавливается с помощью следующих параметров:

STORAGE (INITIAL целое_число {К | М} NEXT целое_число {К | М})

PCTINCREASE целое_число

MINEXTENTS целое_число MAXEXTENTS целое_число

FREELISTS целое_число FREELIST GROUPS целое_число)

где:

INITIAL – размер начального экстента,

NEXT – количество экстентов, выделяемых при создании объекта,

PCTINCREASE – приращение размера каждого следующего экстента,

MINEXTENTS – минимальное количество экстентов (по умолчанию для таблиц 1, для индексов – 2),

MAXEXTENTS – максимальное количество экстентов (по умолчанию не ограничено),

FREELISTS – число списков свободных участков,

FREELIST GROUPS – число групп списков свободных участков.

Параметры распределения памяти в Oracle

Параметр PCTFREE

Параметр PCTFREE устанавливает процент памяти блока, резервируемой (оставляемой свободной) для возможных обновлений строк, уже содержащихся в блоке.

Параметр PCTUSED

После того, как блок данных будет заполнен до процента PCTFREE, этот блок не рассматривается для вставки новых строк до тех пор, пока процент используемой памяти в блоке не упадет ниже параметра PCTUSED. До этого момента свободная память в блоке может использоваться лишь для обновления строк, уже содержащихся в блоке данных.

PCTFREE 0-99 (по умолчанию 10)

PCTUSED 1-99 (по умолчанию 40)

$PCTFREE + PCTUSED \leq 100\%$