Part 1 .Administration

1. Oracle Architecture
2. Oracle SGA (System Global Area)
3. Oracle Background Processes
4. Oracle Startup & Shutdown
5. Control files
6. Redo log files
7. Tablespace & Data files
8. oracle 저장구조

8. oracle 저장구조

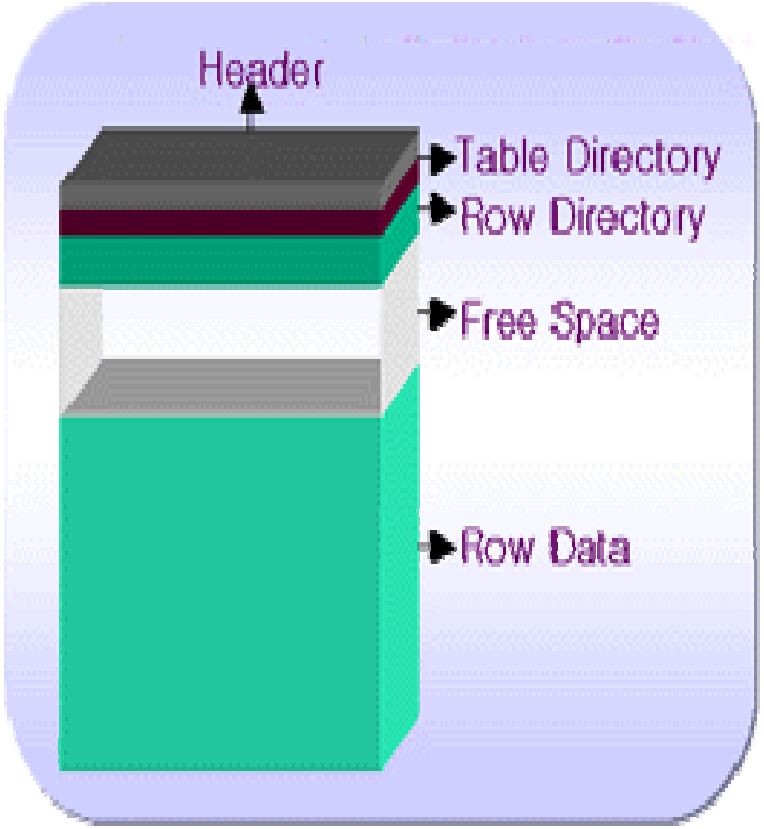
- Oracle Block 개요
  - Oracle에서의 최소 작업 단위
  - 사용자가 입력한 데이터를 하드디스크에 저장하거나 읽어들 일 때 1건씩 처리하는 구조가 아닌, 가장 작은 단위인 block 단위로 작업함
  - DB_BLOCK_SIZE 파라미터로 block 사이즈 조절가능, default는 8K
  - create database 할 때 한 번 지정되면 그 값은 database를 재 생성하기 전에는 변경할 수 없다
  - SQL> show parameter db_block_size 로 조회

8. oracle 저장구조

- Oracle Block 개요
  - 2KB부터 4KB, 8KB, 16KB, 32KB가 제공
  - 짝수 SIZE 권장
  - block 사이즈가 작다면 한번에 담을 수 있는 데이터 양이 작아져 disk I/O가 많아짐
  - block 사이즈가 크면 데이터가 작을 경우 공간이 낭비될 수 있고 wait time이 많이 생길 수 있음
  - Tablespace 생성 시 block size를 기존과 다르게 설정할 수 있으나 먼저 database buffer cache에도 해당 block 사이즈 만큼 공간을 미리 할당해두어야 설정 가능(권고X)

8. oracle 저장구조

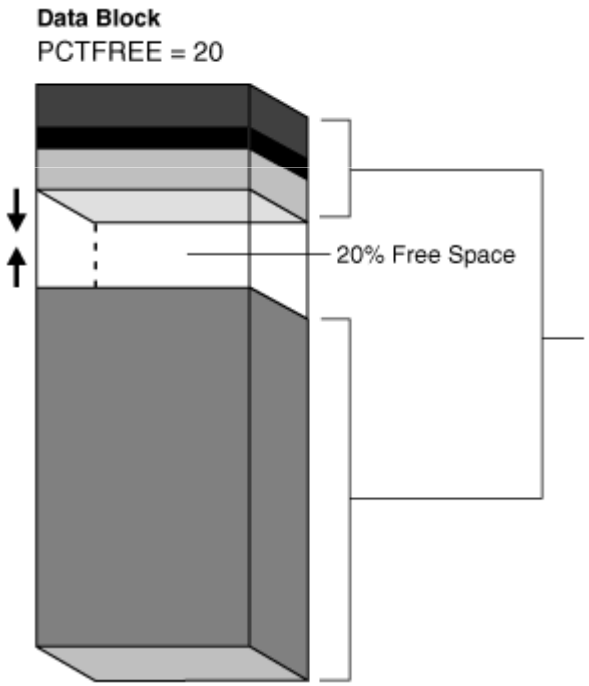
- Oracle Data Block 상세구조



- 블록헤더(Header) : 일반적인 Block의 정보를 가짐 (Block의 위치, Segment의 형태)
- Table Directory : 클러스터에 있는 테이블에 관한 정보를 가짐
- Row Directory : Block내의 Row관련 정보를 가짐
- Free Space : New Row Insert나 Update시 사용 PCTFREE값과 PCTUSED에 의해 결정
- Row Data : 실제 테이블 데이터와 인덱스 데이터가 저장됨

8. oracle 저장구조

- PCTFREE



- 사용 가능한 Block 공간 중에서 데이터 행의 UPDATE시 block의 여유 공간이 없어 다른 block에 저장될 것에 대비하여 확보해 놓는 영역

- PCTFREE의 기본값은 10%

- PCTFREE값에 도달하면 블록은 꽉 찬 것으로 간주하고, 새로운 Row를 추가할 수 없음

8. oracle 저장구조

- PCTFREE
  - 이 공간은 데이터가 insert 되지 않고 오로지 update만을 위해 남겨놓은 공간
  - PCTFREE값이 도달된 후 블록 내에 남은 공간은 기존의 Row가 Update되어 공간이 추가로 필요할 때 사용됨
  - 이 공간을 제외한 나머지 공간이 가득 차게 되면 이 block 은 더 이상 빈 공간이 없는 Dirty Block 상태로 변경
  - PCTFREE와 PCTUSED의 합이 100을 초과하지 않는 범위 내에서 0에서 99까지의 값을 PCTFREE 값으로 사용할 수 있음 (PCTFREE+PCTUSED <= 100)

8. oracle 저장구조

- **PCTFREE가 적을 경우**
  - 기존 테이블 행 갱신에 의한 확장을 위해 적은 공간을 확보
  - update가 빈번히 일어나는 경우 Row Migration이 자주 발생
  - 많은 row가 한 블록에 입력 가능
  - 수정이 적은 세그먼트에 적합

8. oracle 저장구조

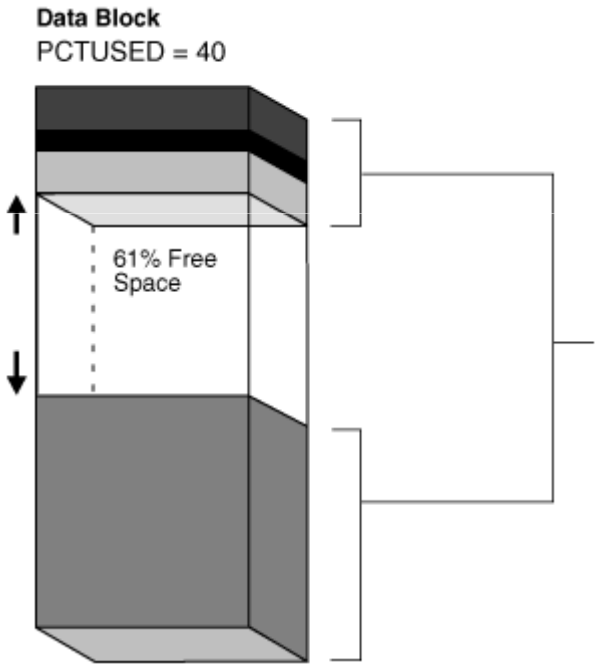
- **PCTFREE가 클 경우**
  - 블록당 적은 row가 입력, 즉 같은 row를 입력 하기 위해서 많은 블록이 필요
  - update시 다른 block으로 이동할 필요가 없으므로 수정 수행 속도가 증가
  - 자주 수정되는 세그먼트에 적합

8. oracle 저장구조

- block내 빈 공간을 관리 하는 과정
  - 비어있는 block의 명단은 Free list에 다 기록되고, 빈 공간이 없는 block은 Dirty list에 기록되는데 데이터 1건 지울 때 마다 Free list와 Dirty list를 업데이트 하는 경우 부하가 매우 큼
  - block에 있는 row가 삭제 되어서 사용량이 줄어 들어도 즉시 이 block을 재사용하지 않고 특정 양이 빌 때까지 기다렸다 재사용 함
  - 지워진 후 남은 용량이 얼마가 되어야 Free block이 되는지를 결정하는 파라미터가 PCTUSED

8. oracle 저장구조

- PCTUSED



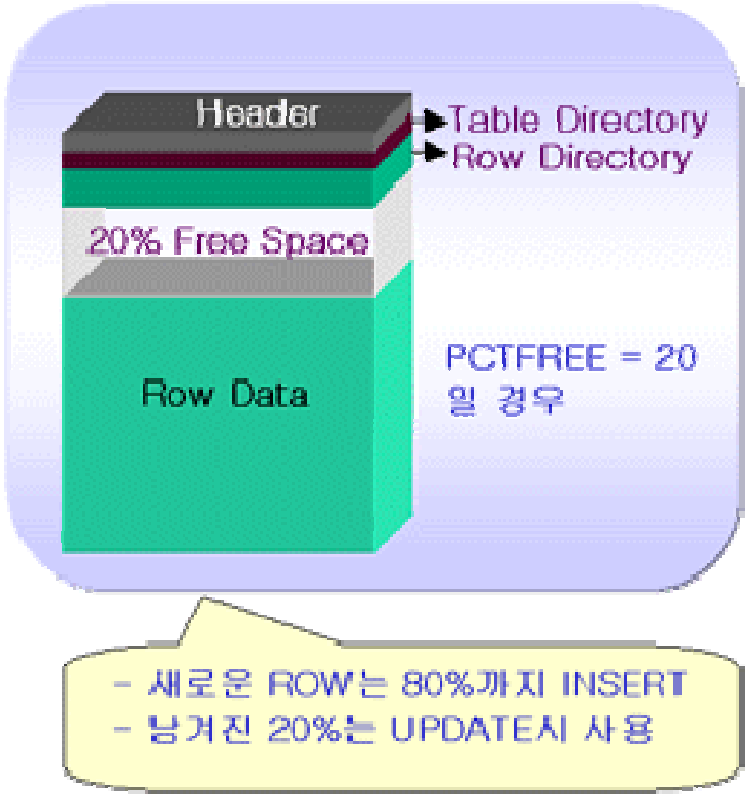
- 계속해서 데이터가 지워져서 빈 공간이 생길 경우 Dirty block에서 Free block으로 변경되는 기준을 의미

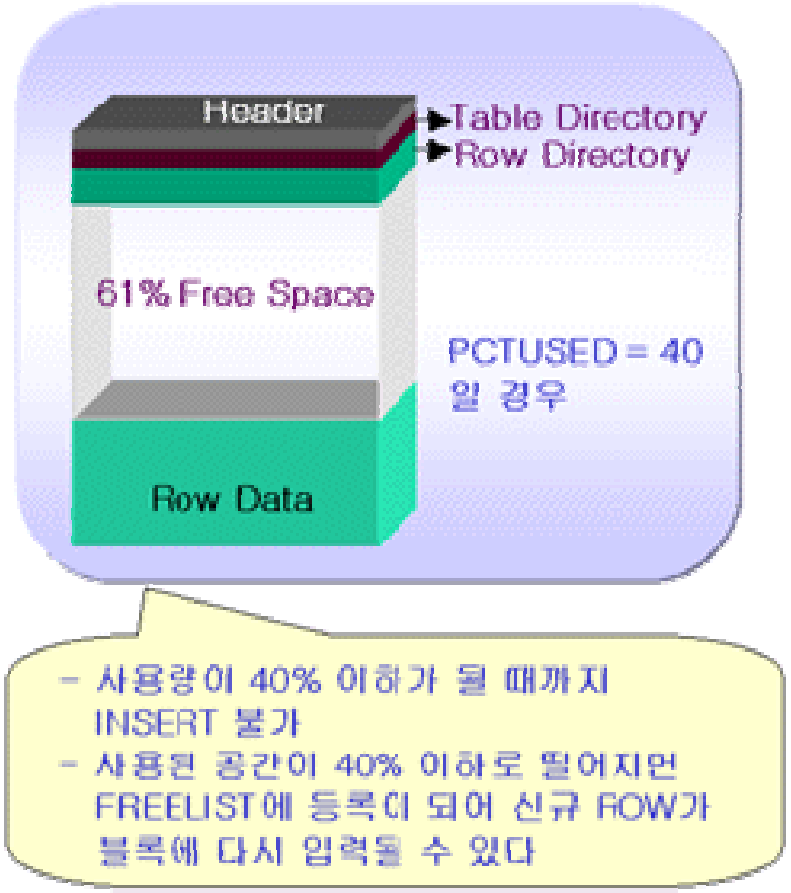
- PCTUSED = 40 : 데이터 블록의 사용 영역이 40%보다 적어지지 않으면 새로운 행을 INSERT할 수 없음

- ASSM(Automatic Segment Space Management)일 경우 사용되지 않음 (default)

8. oracle 저장구조

- PCTFREE와 PCTUSED





8. oracle 저장구조

- **Row Chaining**
  - Row Chaining은 DB_BLOCK_SIZE보다 너무 큰 데이터가 들어왔을 경우, 혹은 남아 있는 block size보다 큰 데이터가 들어올 경우 인접한 다른 블록까지 데이터가 쓰여지는 것
  - 이 데이터 블록은 원래 블록과 연결 되어짐
  - Row Chaining은 블록이 하나의 I/O 작업과 동일한 양을 수행하기 위해 두 개의 I/O를 사용해야 하므로 성능상의 부하를 줌
  - Row Chaining 현상을 줄이기 위해서는 block size를 크게 생성하는 것이 좋음

8. oracle 저장구조

- **Row Migration**
  - UPDATE시 row에 저장된 Data의 양을 증가시켜 더 이상 그 블록에 저장될 수 없을 경우 새로운 Block으로 데이터를 완전히 옮기는데 이를 Row Migration 이라고 함
  - Update시 공간이 부족할 때 발생하며, Data 행을 완전히 채울 수 있는 새로운 Block 에 저장 됨
  - 원래 block에 신규 block의 포인터 주소를 남겨두어 이전 block으로 찾아오는 access도 신규 block을 찾아갈 수 있도록 함
  - PCTFREE값을 많이 주거나 reorg 등의 작업을 통해 줄일 수 있음

8. oracle 저장구조

- **Extent**
  - Extent는 일정한 수의 연속된 oracle block
  - 일정한 수라는 의미는 사용자가 지정 한 값을 의미
  - create table 옵션에 INITIAL EXTENT로 설정
  - INITIAL EXTENT = 64K이면 DB_BLOCK_SIZE=8K일 경우 연속해서 8개의 block이 1개의 extent를 구성
  - 10g 버전부터는 ASSM(Automatic Segment Space Management)이 기본이며, 이때 extent size는 64K임

8. oracle 저장구조

- **Extent 사용 이유**
  - 해당 테이블이 사용할 extent를 미리 할당함으로써 data file 내 곳곳에 분산되어 저장되는 것을 피하기 위함
  - 동일 table의 data가 곳곳의 block에 저장되어 있다면 이를 찾는 시간이 오래 걸려 전체적인 성능 저하 발생

8. oracle 저장구조

- **Extent 특징**
  - Object는 현재 할당된 모든 Extent가 이미 채워진 경우에만 새로운 Extent를 할당
  - 새로운 Extent는 Next Extent 크기만큼 할당(10g 이상부터는 64KB가 기본)
  - Next Extent = Next * (1+Pctincrease/100)
  - Free Extent를 할당하고 관리하는 방식은 DMT와 LMT로 나뉘게 된다

8. oracle 저장구조

- **DICTIONARY MANAGED TABLESPACE(DMT)**
  - 전통적인 방법으로 STORAGE절을 사용하여 EXTENT를 할당
  - Oracle 9i 버전 이후 DICTIONARY MANAGED 방식의 사용을 권장하지 않음
- **LOCALLY MANAGED TABLESPACES(LMT)**
  - CREATE TABLE 절에 STORAGE를 모두 무시
  - TABLESPACE 에서 정한 UNIFORM SIZE 대로 EXTENT 할당
  - Segment를 관리하는 방식에 따라 FLM과 ASSM으로 나뉨

8. oracle 저장구조

- **DICTIONARY MANAGED TABLESPACE(DMT)**

```
SQL> CREATE TABLESPACE local_tbs
     DATAFILE '.....' SIZE 50M
     EXTENT MANAGEMENT DICTIONARY
     DEFAULT STORAGE
     (INITIAL 1M          -- 최초 extent 할당 size
      NEXT 1M             -- 최초 이후 next extent size
      MINEXTENTS 2        -- 생성 할 extent 최소 개수
      MAXEXTENTS 50       -- 생성 할 extent 최대 개수
      PCTINCREASE 0);     -- next 값 증가율
```

8. oracle 저장구조

- **LOCALLY MANAGED TABLESPACES(LMT)**

```
SQL> CREATE TABLESPACE local_tbs
     DATAFILE '.....' SIZE 50M
     EXTENT MANAGEMENT LOCAL  -- 생략시 default
     UNIFORM SIZE 10M;
```

✓ UNIFORM : 테이블스페이스가 동일한 크기의 extent로 구성되도록 지정

8. oracle 저장구조

- **FLM(Free List Management)**
  - Oracle 8i까지의 Segment 관리 기법
  - Segment의 Free Block들은 항상 freelist를 통해 관리됨
  - PCTUSED 아래로 채워진 block들이 freelist로 연결되어 있어서, insert가 필요하면 이 freelist를 segment header에서부터 뒤지면서 블록내의 빈 공간에 insert를 하게 됨

8. oracle 저장구조

- **ASSM(Automatic Segment Space Management)**
  - Oracle 9i부터는 PCTFREE, PCTUSED를 직접 지정하는 기존의 FLM 방법 대신 자동으로 관리하는 방법을 권장
  - segment에 할당된 space를 freelist 대신 bitmap으로 관리
  - ASSM 방법을 이용하여 space를 관리하게 되면 freelist를 타고 다음다음 block을 access하는 대신 Tree 구조의 bitmap 정보를 참고로 적당한 block들을 선택하기 때문에 space에 관한 성능이 훨씬 좋아짐
  - ASSM 방식을 이용하려면 반드시 LOCALLY MANAGED TABLESPACE여야 함

8. oracle 저장구조

- **ASSM(Automatic Segment Space Management)**

```
SQL> CREATE TABLESPACE assm_tbs1
       DATAFILE '.....' SIZE 10M
       EXTENT MANAGEMENT LOCAL
       SEGMENT SPACE MANAGEMENT AUTO;  -- 생략 시 default
```

  - AUTO가 아닌 MANUAL로 지정되게 되면 FLM 방식

  - DBA_TABLESPACES view의 SEGMENT_SPACE_MANAGEMENT column을 통해 AUTO인지 MANUAL방식인지 확인이 가능

8. oracle 저장구조

- Segment
  - Oracle Object 중에서 독자적인 저장 공간을 가지는 것
  - 데이터가 증가되면 실제 증가되는 영역
  - Table, Index, Undo, Temp 등이 대표적인 Segment의 종류
  - 여러 개의 Extent로 구성