

# Oracle DBA 면접 개념 정리

DATE : 2011-04-20 19:10

Writer : 박상수

License : 개인자료

**[ESTENPARK.TISTORY.COM](http://ESTENPARK.TISTORY.COM)**

- 본 문서는 모든 사람에게 열람 및 수정 모두 가능합니다 -

버전	수정일	작성자	변경 내역
1.0	2011/03/02	박상수	면접 족보집 작성
1.5	2011/03/11	박상수	면접 ADMIN 문서 정리
2.0	2011/03/13	박상수	면접 B/R 문서 정리
3.0	2011/04/20	박상수	면접 세부 내용 추가 및 업데이트
3.5	2010/04/20	박상수	면접 SQL 문서 정리

# 목 차

1. Oracle 기본(데이터베이스).....	6
1.1. 데이터베이스란? .....	6
1.2. DBMS의 언어.....	6
1.3. RDBMS 란? .....	7
2. 오라클 SQL .....	8
2.1. 오라클 데이터 타입의 종류를 설명 하세요.....	8
2.2. 오라클 일반 함수에 대해서 설명 하세요. ....	8
2.3. 오라클 그룹 함수에 대해서 설명 하세요. ....	9
2.4. 집합 연산자에 대해서 설명 하세요.....	10
2.5. 조인에 대해서 설명 하세요. ....	10
2.6. 조인(JOIN)의 종류에 대해서 설명 하세요. ....	10
2.7. SUB QUERY에 대해서 설명 하세요. ....	11
2.8. 오라클 데이터 무결성 제약조건에 대해서 설명 하세요.....	11
3. 오라클 설치 관련 .....	12
3.1. 싱글 설치 해보셨나요? 설치부터 패치까지 설명 하세요. ....	12
3.2. 오라클에서 Patch set, PSU, CPU 의 차이점은 무엇인가요?.....	12
3.3. 싱글 설치 완료 후 해야 되는 작업은 무엇이 있습니까? .....	12
3.4. 오라클 업그레이드 작업은 해보셨나요? 어떻게 하는지 설명 하세요?.....	12
3.5. OLTP, OLAP 차이점은 무엇인가요?.....	13
3.6. 커널 매개변수를 변경하는 파일은? .....	13
3.7. RAC 설치해보셨나요? 그럼 설치하는 방법을 설명 하세요? .....	13
3.8. Oracle에서 RAW DEVICE를 사용하는 이유는? .....	14
3.9. 오라클에서 ASM에 대해서 설명 하세요?.....	15
3.10. ASM과 파일시스템의 차이점 .....	15
3.11. Opatch 사용 방법은?.....	15
3.12. 오라클 버전의 차이점(9i, 10g, 11g) .....	16
3.13. RAC 캐시퓨전이란? .....	16
3.14. RAC 에서 OCR, Voting disk에 대해서 설명 하세요.....	16
3.15. RAC IP관련하여 자세히 설명하시오?.....	17
4. 오라클 ADMIN 관련 .....	18

4.1. 오라클 구조에 대해서 설명 하세요.....	18
4.2. PGA의 구성요소 및 각 구성요소의 특징을 설명하세요.....	18
4.3. SGA, Instance, Database의 차이점은 무엇인가요?.....	18
4.4. SGA의 정의와 구성요소를 설명 하시기 바랍니다.....	18
4.5. Shared Pool에 대해서 설명 하시기 바랍니다.....	18
4.6. LRU 알고리즘.....	19
4.7. ASMM이란 ?.....	19
4.8. Data Buffer Cache 대해서 설명 하세요.....	19
4.9. Redo Log Buffer에 대해서 설명 하세요.....	20
4.10. 오라클 Background Process의 종류를 말하세요.....	20
4.11. 오라클 Background Process의 각각의 역할을 설명 하세요.....	20
4.12. 오라클 시작과 종료에 대해서 절차적으로 설명 하세요.....	21
4.13. UNDO Retention은 무엇입니까?.....	22
4.14. Redo와 Undo의 차이점.....	22
4.15. Index 사용이유?.....	22
4.16. Index의 종류와 그 기능.....	22
4.17. Index SPLIT 대한 정의 및 발생원인, 대해 설명 하세요.....	23
4.18. FRA 란.....	24
4.19. ADDM이란?.....	24
4.20. Oracle 리오그(Reorganization) 작업은?.....	24
4.21. Select 의 문장이 실행되는 과정?.....	25
4.22. Update(DML)의 문장이 실행되는 과정.....	25
4.23. Commit 실행 되는 과정.....	25
4.24. 오라클의 커서는 무엇인가?.....	26
4.25. 오라클 트리거란?.....	26
4.26. 오라클 동의어(Synonym)란?.....	26
4.27. 오라클 시퀀스(Sequence)란?.....	26
4.28. 익스텐트 관리 방법의 2가지를 설명하세요.....	27
4.29. 세그먼트 공간 관리 방법의 2가지를 설명하세요.....	27
4.30. 정렬 절차에 대해서 설명하세요?.....	27
4.31. 래치((Latch)와 락(Lock).....	28
4.32. Incremental Checkpoint와 Normal Checkpoint의 차이점?.....	28
5. 오라클 백업과 복구.....	30
5.1. Archive log 모드와 No Archive log 차이점.....	30
5.2. Archive Hang 원인과 해결 방법은?.....	30

5.3. Oracle 닫힌 백업(cold backup), 열린 백업(hot backup)은 무엇인가요?.....	31
5.4. 데이터 파일 복구 .....	31
5.5. 그 외 복구 방법은 “실전 오라클 백업과 복구 - 서진수 지음” 책을 이용하여 확인 하시 기 바랍니다. 경우의 수가 많아서 문서로 기록하기에는 범위가 너무 많습니다. ....	35
5.6. Export/Import 란 .....	35
5.7. Clone DB 만드는 과정을 설명하세요? .....	36
5.8. Flashback란? .....	36
5.9. Recovery Manger .....	36
6. 기타 .....	38
7. 도와주신 분들 .....	39

# 1. Oracle 기본(데이터베이스)

## 1.1. 데이터베이스란?

- 방대한 양의 데이터를 저장하고 관리하면서 데이터베이스 시스템의 중심에 있게 되었습니다. 대용량 데이터베이스 시스템들은 다양한 모습을 갖추게 되면서 점차 DBMS(Database Management System)은 지속적으로 발전해 가고 그 역할을 확대시키고 있습니다.
- 1960년대에 계층형 DBMS를 시작으로 1970년대에 접어들면서 처음으로 RDBMS가 상용화 되기 시작 하였으며, 1990년대에 들어서는 좀 더 진보된 개념으로서 OODBMS와 ORDBMS가 등장 하게 됩니다. 그러나 현재까지 정보시스템에서 가장 일반적으로 널리 사용되고 있는 것은 RDBMS입니다. 최근에는 MMDBMS(Main Memory Database Management System)으로 데이터베이스의 일부 또는 전부를 메인 메모리 상에서 관리하는 것인데, 이는 디스크에 대한 접근 없이 직접 메인 메모리접근을 통해 데이터를 관리함으로써 고성능 트랜잭션 처리를 가능하게 합니다. Virtual Memory와 Main Memory를 혼동하는 경우가 많은데 Virtual Memory는 자료를 처리하는 데에 있어서 실제의 메모리를 사용하는 것이 아니고 가상 메모리를 사용하는 것으로 IBM에서 사용하는 독특한 자료처리 방식입니다. 현재 메인 메모리 데이터베이스(MMDB), 주기억장치 상주 데이터베이스(MRDB), 또는 인-메모리 데이터베이스(IMDB) 등의 이름으로 불리는데 데이터베이스를 메모리에 위치시켜 놓고 사용한다는 동일한 개념을 내포한 이음동언어 들입니다.
- 참고자료 :
  - <http://www.dbguide.net/db.db?cmd=view&boardUid=13875&boardConfigUid=9&categoryUid=216&boardIdx=84&boardStep=1>
  - 초보자를 위한 오라클 10g 18p

## 1.2. DBMS의 언어

- 데이터 정의어(DLL, Data Definition Language) : 데이터 정의를 통해서 데이터베이스를 구성하는 오브젝트가 정의되며 데이터베이스의 메타 데이터로 저장됩니다.
  - CREATE, ALTER, DROP, TRUNCATE
- 데이터 제어어(DCL, Data Control Language) : 데이터베이스에 저장된 데이터를 올바르게 유지하기 위해 사용되며 데이터 무결성 또는 보안에 대한 작업을 수행하게 됩니다.
  - GRANT, REVOKE
- 데이터 조작어(DML, Data Manipulation Language) : 데이터베이스에 데이터를 입력, 수정 및 삭제 등의 작업을 수행하기 위해서는 각각의 작업에 맞는 DML을 이용해야 합니다.
  - UPDATE, INSERT, DELETE, MERGE
- 데이터 검색(Query) : 데이터베이스에 저장되어 있는 데이터를 조회하기 위해 사용하는 SQL 언어입니다.
- 참고 자료 : <http://estepark.tistory.com/289>

### 1.3. RDBMS 란?

- RDBMS(Relational Database Management System)는 인포믹스와 IBM의 DB2가 상용화의 모습을 갖추었으나 현재는 그 외에도 많은 종류의 RDBMS가 사용되고 되며, 대표적으로 오라클, MS SQL, 사이베이스, DB2 및 인포믹스가 있습니다.
- 관계 데이터 모델은 테이블을 기반으로 하기 때문에 모델이 간단하여 이해하기 쉽습니다. 그러나 복잡한 오브젝트들에 대한 유지와 변경이 어렵고 다차원적인 데이터들을 표현하기가 쉽지 않았습니다. 그래서 이를 보완한 개념으로 객체 지향 DBMS가 등장하였습니다.

## 2. 오라클 SQL

### 2.1. 오라클 데이터 타입의 종류를 설명 하세요.

- Char(size) : size 크기의 고정길이 문자 데이터 타입
- Varchar(size) : size 크기의 가변길이 문자 데이터 타입
- Nvarchar(size) : 국가별 문자집합에 대한 size 크기의 문자 또는 바이트의 가변길이
- Number(p,s) : 정밀도(P)와 스케일(S)로 표현되는 숫자 데이터 타입
- DATE : 날짜 형식을 저장하기 위한 데이터 타입
- ROWID : 테이블내 행의 고유주소를 가지는 64진수 문자타입, 해당 6바이트 또는 10 바이트
- BLOB : 대용량의 바이너리 데이터를 저장하기 위한 데이터 타입으로 최대크기는 4GB
- CLOB : 대용량의 텍스트 데이터를 저장하기 위한 데이터 타입으로 최대크기는 4GB
- BFILE : 대용량의 바이너리 데이터를 파일 형태로 저장하기 위한 데이터 타입 최대크기 4GB
- TIMESTAMP(n) : DATE 데이터 타입의 확장된 형태 n은 Milli second 자리수로 최대 9자리까지 가능
- INTERVAR YEAR TO MONTH : 년과 월을 이용하여 기간을 저장
- INTERVAR DAY TO SECOND : 일, 시, 분, 초를 이용하여 기간을 저장 두 날짜값의 정확한 차이를 표현 함

### 2.2. 오라클 일반 함수에 대해서 설명 하세요.

- 대소문자 변환함수
  - INITCAP : 문자열의 첫 번째 문자만 대문자로 변환
  - LOWER : 문자열 전체를 소문자로 변환
  - UPPER : 문자열 전체를 대문자로 변환
- 문자열 길이 반환 함수
  - LENGTH : 문자열의 길이를 반환
  - LENGTHB : 문자열의 바이트 수를 반환
- 문자 조작 함수
  - CONCAT : 두 문자열을 결합, || 와 동일한 구성을 함
  - SUBSTR : 특정 문자 또는 문자열 일부를 추출
  - INSTR : 특정 문자가 출현하는 첫 번째 위치 반환
  - LPAD : 오른쪽 정렬 후 왼쪽에 지정한 문자 삽입
  - RPAD : 왼쪽 정렬 후 오른쪽에 지정한 문자 삽입
  - LTRIM : 왼쪽의 지정 문자를 삭제
  - RTRIM : 오른쪽의 지정 문자를 삭제
- 숫자 함수
  - ROUND : 지정 한 자리 이하에서 반올림



- TRUNC : 지정 한 자리 이하에서 절삭
- MOD : m을 n으로 나눈 나머지 값
- CEIL : 지정한 값 보다 큰 수 중에서 가장 작은 정수
- FLOOR : 지정한 값 보다 작은 수 중에서 가장 큰 정수
- 날짜 계산
  - 날짜 + 숫자 = 날짜 : 날짜에 일수를 가산하여 날짜 계산
  - 날짜 - 숫자 = 날짜 : 날짜에 일수를 감산하여 날짜 계산
  - 날짜 - 날짜 = 일수 : 날짜와 날짜를 감산하여 일수 계산
  - 날짜 + 숫자 / 24 = 날짜 : 날짜에 시간을 가산하여 날짜를 계산
- 날짜 함수
  - SYSDATE : 시스템 현재 날짜(결과 = 날짜)
  - MONTH\_BETWEEN : 날짜와 날짜 사이의 개월을 개산(결과 = 숫자)
  - ADD\_MONTHS : 날짜에 개월을 더한 날짜 계산(결과 = 날짜)
  - NETX\_DAY : 날짜 후의 첫 월요일 날짜를 계산(결과 = 날짜)
  - LAST\_DAY : 월의 마지막 날짜를 계산(결과 = 날짜)
  - ROUND : 날짜를 반올림(결과 = 날짜)
  - TRUNC : 날짜를 절삭(결과 = 날짜)
- 데이터 타입의 변환(묵시적 변환)
  - NUMBER 타입을 VARCHAR2, CHAR를 변환 할 경우 VARCHAR2, CHAR은 NUMBER로 변환
  - VARCHAR2, CHAR 타입을 NUMBER를 변환 할 경우 NUMBER은 NUMBER로 변환
  - (주의사항) 문자 타입의 숫자 타입 변환은 문자열이 숫자일 경우에만 가능 합니다.
- 데이터 타입의 변환(명시적 변환)
  - TO\_CHAR : 숫자, 날짜를 문자 타입으로 변환
  - TO\_NUMBER : 문자열을 숫자 타입으로 변환
  - TO\_DATE : 문자열을 날짜 타입으로 변환

### 2.3. 오라클 그룹 함수에 대해서 설명 하세요.

- COUNT : 행의 개수 출력
- MAX : NULL을 제외한 모든 행에서 최대값 출력
- MIN : NULL을 제외한 모든 행에서 최소값 출력
- SUM : NULL을 제외한 모든 행의 합계
- AVG : NULL을 제외한 모든 행의 평균 값
- STDDEV : NULL을 제외한 모든 행의 표준편차
- VARIANCE : NULL을 제외한 모든 행의 분산 값
- GROUPING : 해당 칼럼이 그룹에 사용되었는지 여부를 1 또는 0으로 변환
- GROUPING SET : 한번의 질의로 여러 개의 그룹화 기능

## 2.4. 집합 연산자에 대해서 설명 하세요.

- UNION : 합집합을 구해줍니다.(중복 자료 배제하기 때문에 성능 좋음)
- UNION ALL : 합집합을 구해줍니다.(중복 자료를 포함 하기 때문에 성능 나쁨)
- MINUS : 차집합을 구해줍니다.
- INTERSECT : 교집합을 구해줍니다.

## 2.5. 조인에 대해서 설명 하세요.

- 조인(JOIN)은 두 개 이상의 테이블이나 DATASET을 서로 연결하는 데이터를 검색 하고자 할 때 사용하는 방법입니다. FROM 절에 조인할 테이블들을 기술하고, 조건 절에서 테이블간의 연결이 이루어질 조건을 서술하면 됩니다.
- 조인의 종류를 보면, CARTESIAN PRODUCT, EQUI JOIN, NON-EQUI JOIN, OUTER JOIN, SELF JOIN 등으로 나뉘져있습니다.

## 2.6. 조인(JOIN)의 종류에 대해서 설명 하세요.

- 카티션 곱(Cartesian Product)
  - 두 개 이상의 테이블에 대해 연결 가능한 행을 모두 결합하는 조인 방법이며, WHERE 절에서 조인 조건절을 생략하거나 조인 조건을 잘못 설정하여 양쪽 테이블을 연결하여 조건에 만족하는 행이 하나도 없는 경우에 발생 합니다.
  - 예) SELECT \* FROM DEPT A, EMP B
- EQUI JOIN
  - SQL 문에서 가장 많이 사용되는 조인으로 조인 대상 테이블에서 공통 칼럼을 '=' 비교를 통해 같은 값을 갖는 행에 대해서 연결하여 결과를 생성하는 조인 방법입니다.
  - 예) SELECT A.name FROM DEPT A, EMP B WHERE A.code = B.code
- NON-EQUI JOIN
  - 조인 조건에서 '<','>', BETWEEN a AND b'와 같이 '=' 조건이 아닌 다른 종류의 연산자를 사용하는 조인 방법입니다.
  - 예) SELECT A.name FROM DEPT A, EMP B WHERE A.code > B.code
- OUTER JOIN
  - 두 개 이상의 테이블 조인 시 한쪽 테이블의 행에 대해 다른 쪽 테이블에 일치하는 행이 없더라도 다른 쪽 테이블의 행을 NULL로 하여 행을 Return하는 것이 OUTER JOIN 입니다.연산자로는 (+)기호를 사용합니다.
  - 예) SELECT A.name FROM DEPT A, EMP B WHERE A.code = B.code(+)
- SELF JOIN
  - 하나의 테이블에서 두 개의 칼럼을 연결하여 EQUI JOIN을 하는 방법입니다.
  - 예) SELECT A.name FROM DEPT A, DEPT B WHERE A.code = B.code

## 2.7. SUB QUERY에 대해서 설명 하세요.

- 하나의 SQL 명령문의 처리 결과를 다른 SQL 명령문에 전달하기 위해서 두 개 이상의 SQL문을 하나의 SQL 명령문으로 연결 합니다. 서브 쿼리를 포함한 SQL 명령문을 메인 쿼리 라고 하며, 서브 쿼리는 SELECT 문의 시작과 끝에 라이트, 레프트 퍼렌씨시스(Parenthesis) '( )'를 묶어서 메인 쿼리 구분 합니다.
- 단일 행 서브쿼리 : 서브쿼리 검색 결과로 하나의 행만 메인 쿼리로 반환 합니다.
- 다중 행 서브쿼리 : 하나 이상의 행을 메인 쿼리로 반환 합니다.
- 다중 칼럼 서브쿼리 : 하나 이상의 칼럼을 메인 쿼리로 반환 합니다.
- 상호 관련 서브쿼리 : 서브쿼리와 메인 쿼리간에 결과값을 서로 주고 받는 식으로 처리되는 서브쿼리입니다.

## 2.8. 오라클 데이터 무결성 제약조건에 대해서 설명 하세요.

- NOT NULL : 해당 칼럼 값은 NULL을 포함 할 수 없습니다.
- Unique Key(고유키) : 테이블에서 해당 칼럼 값은 반드시 유일 해야 합니다.
- Primary Key(기본키) : 해당 칼럼 값은 반드시 존재해야 하며, 유일 해야 합니다. Unique Key, NOT NULL 제약 조건이 결합된 형태
- 참조
  - 한 테이블의 칼럼 값이 자식 또는 다른 테이블의 칼럼 값 중 하나와 일치시키기 위한 제약 조건
  - 자식 테이블 : 다른 테이블의 값을 참조하는 테이블
  - 외래키(Foreign Key) : 부모 테이블의 값을 참조하는 자식 테이블의 칼럼
  - 부모 테이블(Reference Key) : 다른 테이블에 의해 참조되는 테이블
  - 참조키(Foreign Key) : 자식 테이블에서 참조하는 부모 테이블의 칼럼
- CHECK : 칼럼에서 허용 가능한 데이터의 범위나 조건을 지정하기 위한 제약 조건

## 3. 오라클 설치 관련

### 3.1. 싱글 설치 해보셨나요? 설치부터 패치까지 설명 하세요.

- OS 설치(linux, unix)
- OS 패치
- 오라클 설치를 위한 환경 설정
- 엔진 설치
- 오라클 엔진 패치
- 리스너 구성
- 데이터베이스 생성 (수동/자동으로 하는 방식이 있습니다.)

### 3.2. 오라클에서 Patch set, PSU, CPU 의 차이점은 무엇인가요?

- 오라클에서 Patch Set은 릴리즈 버전의 통합 패치 파일입니다,
- PSU(Patch Set Update)는 오라클에서 정기적으로 권고하는 패치로써 단일 패치를 통합 한 것을 의미 합니다. 예를 들어서 분기별로 총 4번 제공 합니다.(10.2.0.2.1~4)
- CPU(Critical, Patch Update)는 오라클 제품의 보안 문제를 해결하기 위한 패치 입니다. 역시 PSU와 같이 분기별로 총 4번 제공 합니다.

### 3.3. 싱글 설치 완료 후 해야 되는 작업은 무엇이 있습니까?

- 데이터 파일, 컨트롤 파일, 리두 로그파일 이중화 작업
- 테이블 명세서를 통해서 유저, 테이블스페이스, 테이블, 인덱스, 트리거 등등 작업

### 3.4. 오라클 업그레이드 작업은 해보셨나요? 어떻게 하는지 설명 하세요?

- 이전 버전 구성요소를 확인 합니다. (서버 종류, 메모리, CPU, hostname 등등)
- 업그레이드 할 버전의 엔진을 설치 합니다.(10g Profile)
- 업그레이드 할 버전의 Companion을 설치 합니다.
- 업그레이드 할 버전의 Patch Set를 설치 합니다.
- 업그레이드 한 버전에서 이전버전 체크 할 사항
  - utlu102i.sql로 체크 합니다.(9i Profile) - /oracle/product/102/rdbms/admin/utlu112i.sql
  - connect role을 가진 user확인
  - DB 링크 생성
  - Timezone, national character set check, 통계 수집, Invalid Object check

- Dictionary의 corruption을 체크 하는 스크립트 작성
- MVIEW / SNAPSHOT refresh 여부 확인
- 리스너 종료, 백업 중인 파일 확인, 완료 되지 않은 트랜잭션 검색
- OS상 Cron job확인
- System tablespace, temporary tablespace 의 기본 유저 확인
- aud\$가 system tablespace 안에 있는지 확인
- controlfile 경로 확인
- XDB.MIGR9202STATUS 테이블 존재 유무 확인
- 이전 DB 종료 후 Full Backup
- 이전 DB에서 사용하던 INIT[SID].ora 파일 업그레이드 DB로 복사
- 넷 서비스 설정 파일 모두 복사
- oratab 수정
- 업그레이드 시작
  - startup upgrade
  - sysaux 테이블 스페이스 생성
  - @?/rdbms/admin/catupgrd.sql
  - @?/rdbms/admin/utlu102s.sql
  - DB 종료 후 init[SID].ora 파일 최종 수정
  - listener.ora 파일 수정
  - DB 시작 및 alert 로그 확인
  - 리스너/companion 확인

### 3.5. OLTP, OLAP 차이점은 무엇인가요?

- OLTP(Online Transaction Proccession) : 다수의 사용자가 실시간으로 데이터베이스의 데이터를 생성하거나 조회하는 등의 단위 작업을 처리하는 방식(B\*Tree 인덱스 적합)
- OLAP(Online Analytical Processing) : OLTP와 상대되는 개념으로 쌓여있는 데이터를 분석 다차원적으로 활용 하는 방식(BITMAP 인덱스 적합)

### 3.6. 커널 매개변수를 변경하는 파일은?

- Solaris : /etc/system  
set semsys:seminfosemnmi=100
- Linux : /etc/sysctl.conf

### 3.7. RAC 설치해보셨나요? 그럼 설치하는 방법을 설명 하세요?

- RAC 설치 는 기본적으로 하드웨어 부분을 준비해야 합니다. 네트워크 NAT, host only, 하드디스크는 ASM경우 기본적으로 24개가 필요하고 로우 디바이스의 경우에는 하드디스크를 하나만 생성하여

LVM으로 파일시스템을 관리 해도 됩니다.

- 노드 1에서 OS를 설치 합니다.
- 노드 1에서 환경 설정을 합니다.
  - 사용하지 않는 서비스 중단
  - Run level 변경(기본적으로 텍스트 모드로 나오게 하기 위해서 설정 합니다.)
  - 네트워크 설정 변경(NAT, Host Only IP, Netmask, Route 등등 설정)
  - /etc/hosts 설정(Public IP, Private IP, Public Virtual IP)
  - 커널 매개변수 변경 및 추가
  - Shell Limit 설정(제한)
  - /etc/profile 설정
  - /etc/modprobe.conf 설정하여 커널 기반 타이머 설정(RAC1 RAC2 시간 동기화 작업)
  - /etc/rc.local 서버가 재 부팅 되면 시간을 동기화 하기 위한 방법
  - Package 확인
  - Oracle User, Group 생성 및 설정
  - Oracle 계정에서 .bash\_profile(환경설정 파일은 사용하는 셸에 따라 달라집니다.)
  - 디스크 파티션 설정(LVM)을 하여 RAW 디바이스 생성
  - /etc/permissions.d/50-udev.permissions 파일에서 raw의 root 권한을 Oracle로 변경
  - shutdown (종료)
- 노드 1을 노드 2로 복제
- SSH-KEYGEN 설정
- Oracle Clusterware 설치 및 패치
- Oracle 엔진 설치 및 패치
- Oracle 리스너 구성
- ASM 일 경우 엔진 설치 파일에서 configure automatic Storage Management(ASM)을 선택 한 후 패치를 합니다.
- Oracle Database 생성

### 3.8. Oracle에서 RAW DEVICE를 사용하는 이유는?

- raw device는 character device driver를 사용하는 HW DEVICE 입니다.
- 장점
  - RAW Device는 Kernel에 의해 Buffering 되는 것이 아니라 USER Buffer와 Device 간에 직접 DATA가 전송 되므로 Oracle Disk I/O 성능이 향상되고 CPU OVERHEAD가 감소 됩니다.
  - OS 파일 시스템의 OVERHEAD를 피할 수 있습니다.
  - OS Buffer 사이즈를 줄일 수 있습니다.
- 단점
  - 백업이 복잡합니다.

- 오라클에 의해 사용된 RAW Partition은 CYLINDER 0에서 시작하면 안됩니다.
- 정리하자면 파일 시스템을 사용할 경우 디스크에서 I/O는 느린 반면에 RAW DEVICE는 메모리에서 I/O를 사용하기 때문에 속도가 훨씬 빨라집니다.

<http://blog.naver.com/PostView.nhn?blogId=opening2&logNo=20040509787&redirect=Dlog&widgetTypeCall=true>

### 3.9. 오라클에서 ASM에 대해서 설명 하세요.?

- ASM(Automatic Storage Management)은 데이터베이스에서 사용하는 모든 파일(Controlfile, Archive log file, Redo log File, DataDump File, DataFile, SPFILE 등등)에 대해 자동저장공간 관리를 제공 합니다.
- ASM은 기존의 데이터베이스에서 사용하던 파일 관리 개념과 많이 다릅니다. 특히 백업은 RMAN을 통해 할 것을 권장합니다. 그 외에 Archive log나 Redo log 도 ASM에 저장된다면 RMAN 이용해서 백업을 받아야 합니다.
- ASM의 혜택
  - 효율적인 디스크 자원의 관리
  - 자원의 물리적 장애에 대한 관리 향상
  - 디스크 I/O의 효율적인 분산
  - 효율적인 업무에 대한 집중
- <http://blog.naver.com/PostView.nhn?blogId=dangtong76&logNo=140042654289&parentCategoryNo=24&viewDate=&currentPage=1&listtype=0>

### 3.10. ASM과 파일시스템의 차이점

- 데이터베이스 설치
  - 데이터베이스 설치 시 볼륨 매니저를 사용하여 볼륨 그룹을 생성하거나, 파일시스템을 인스톨 하는 작업이 없습니다.
- 디스크 추가
  - 디스크 추가 시 ASM 인스턴스로 접속하여 디스크를 추가하는 SQL만 실행하면 모든 추가적인 작업들은 오라클이 수행 합니다. 반면에 파일시스템의 경우에는 디스크 추가, 볼륨 매니저를 통한 그룹 생성, 파일시스템 생성, 데이터를 추가된 공간으로 이동, I/O 튜닝이 있습니다.
- 디스크 튜닝
  - 무엇보다 ASM의 강력한 기능은 I/O 튜닝에 있다고 할 수 있습니다. 최적의 성능을 유지 시켜 줄 뿐만 아니라 디스크 추가해도 여유공간만 유지해주면 자동으로 튜닝을 합니다.

### 3.11. Opatch 사용 방법은?

- opatch lsinventory : 현재 패치를 확인 합니다.
- opatch apply : 패치를 적용 합니다.(반드시 패치 디렉토리에서 적용 합니다.)

- <http://estepark.tistory.com/search/opatch>

### 3.12. 오라클 버전의 차이점(9i, 10g, 11g)

- 9i에서 10g는 전반적으로 보강되면서 ASMM, ASSM 모드가 변경 되었습니다.
  - ASMM(Automatic Shared Memory Management) 오라클 10g 부터 새로운 기능으로 오라클이 업무 부하를 판단하여 공유 풀, 기본 데이터 버퍼 캐쉬, 대형 풀 및 자바 풀의 크기를 조정하는 기능입니다.
  - ASSM(Automatic Segment Space Management) 자동 세그먼트 공간 관리 방식은 세그먼트를 구성하는 각 익스텐트의 첫 번째 데이터 Block인 익스텐트 헤더에서 비트맵으로 여유 공간을 가지는 데이터 Block을 관리하는 방식입니다.
- 10g에서 11g는 기술적으로 보강된 것이 많지는 않지만 temporary tablespace를 그룹으로 묶어서 수많은 사용자가 분산해서 사용 할 수 있습니다. 특히 Database replay 기능입니다.  
Database Replay 기능은 마치 데이터베이스를 위한 DVD 레코더처럼 동작 합니다. Database Replay 는 SQL 레벨 하단에서 발생하는 모든 데이터베이스 액티비티를 바이너리로 포맷하여 저장하고 데이터베이스 또한 다른 데이터베이스에서 재생할 수 있게 합니다.  
주로 OS 업그레이드(동일한 커널 매계변수를 이용 가능 등등), 패치 적용, 디버깅, 오브젝트 변경, 데이터베이스 업그레이드, 플랫폼 변경, RAC 구성으로의 전환 이종에서 RAC 구성으로 전환은 정말 획기적입니다. 왜냐하면 단일 인스턴스 환경에서 RAC 환경으로 전환하고자 할 때는 워크로드를 캡처해서 RAC 데이터베이스를 재생해보는 수밖에 없습니다.

### 3.13. RAC 캐시퓨전이란?

- OPS에서는 한 노드에서 다른 노드로 데이터를 요청할 경우 해당 데이터를 먼저 디스크에 기록 한 다음에 요청 노드에서 데이터를 읽을 수 있었습니다.(RAC Ping) 캐시 퓨전은 데이터 잠금 상태를 거치지 않고 바로 전송 할 수 있습니다. 이처럼 Oracle Clustering Solution은 노드간 캐시의 일관성을 보장하기 위한 서버간의 통신 방식을 디스크를 이용한 방식에서 초고속 인터커넥트를 이용한 캐시 퓨전(Cache Fusion)으로 변경하면서 획기적인 발전을 가져오게 되었습니다.
- 실질적으로 캐시퓨전에 관련된 작업을 하는 것은 각각의 instance의 process 중에서 Global Cache Service(GCS)를 담당하는 LMS의 역할이 가장 크다고 할 수 있습니다.
- <http://www.gurubee.net/pages/viewpage.action?pageId=4948902> 꼭 읽어보시길 바랍니다.

### 3.14. RAC 에서 OCR, Voting disk에 대해서 설명 하세요.

- OCR : RAC를 구성하는 정보를 저장하는 저장소라는 것이 있습니다. 이러한 OCR정보는 RAC 환경에서 매우 중요한 관리 항목으로 주기적인 백업을 받아 두어야 합니다. 기본적으로 OCR 백업은 4시간 마다 자동으로 백업이 이루어 지며, 비상 상황을 대비하여 3벌의 백업을 자동으로 유지 관리 합니다. OCR은 root 소유로 되어 있으며, Oracle Cluster Repository의 약자 그대로 Cluster의 정보를 담고 있습니다.



- Voting : Oracle(설치시 UID) 소유로 되어 있으며, node를 제거할지 검사하는 용도로 사용됩니다. 다시 말해서 Voting Disk는 각 노드의 status를 확인하기 위한 용도로 사용 합니다.

### 3.15. RAC IP관련하여 자세히 설명하시오?

- Public IP : 관리용 IP(DBA가 접속)
- Private IP : 노드간 정보 교환(외부에서 접속 할 수 없음)
- Virtual IP : CRS가 서비스용으로 사용

## 4. 오라클 ADMIN 관련

### 4.1. 오라클 구조에 대해서 설명 하세요.

- 오라클의 구조를 큰 의미로 정의 하자면 오라클 넷 서비스(리스너), 인스턴스, 데이터베이스로 구성 되어 있습니다.

### 4.2. PGA의 구성요소 및 각 구성요소의 특징을 설명하세요.

- PGA(Program Global Area)는 데이터베이스에 접속하는 서버 프로세스가 독자적으로 사용하는 오라클 메모리 영역이라고 할 수 있습니다.
- 구체적인 구조를 보면 정렬공간(Soft Area), 세션 정보(Session Information), 커서 상태 정보(Cursor State), 변수 저장 공간(Stack Space) 있습니다. (각각에 대한 정리는 초보자를 위한 오라클 10g 96P를 참조하세요.)

### 4.3. SGA, Instance, Database의 차이점은 무엇인가요?

- SGA는 오라클이 데이터를 읽거나 변경하기 위해 사용하는 공용 메모리 영역 이고 Instance는 SGA 메모리 영역과 백그라운드 프로세스를 합쳐서 Database Instance라고 합니다. Database는 datafile, redo log file, controlfile, password file등등을 의미 합니다.

### 4.4. SGA의 정의와 구성요소를 설명 하시기 바랍니다.

- SGA(System Global Area)는 오라클이 데이터를 읽거나 변경하기 위해 사용하는 공용메모리 영역입니다.
- SGA 구조에는 Shared Pool, Data Buffer Cache, Redo Log Buffer, Large Pool, Java Pool로 구성되어 있습니다.

### 4.5. Shared Pool에 대해서 설명 하시기 바랍니다.

- Shared Pool은 기본적으로 고정 영역과 정적 영역으로 구분 되는데 고정 영역은 데이터베이스 OPEN시 Fixed Size를 의미 합니다. 파라미터 설정 값을 자동으로 할당하며 사용자가 임의로 지정 할 수 없습니다. 정적 영역은 Library Cache, Data Dictionary Cache(또는 Row Cache)라고 부른다. Library Cache는 SQL을 실행하게 되면 오라클 내부적으로 사용하는 SQL에 대한 분석 정보(parse tree) 및 실행 계획(Execution Plan)이 저장 됩니다.  
Data Dictionary Cache의 정보는 SYSTEM Tablespace에 있으며, 테이블, 인덱스, 함수, 및 트리거 등등 오라클 오브젝트 정보 및 권한 등의 정보가 저장 됩니다.
- 구분 분석(Parse)에서 SQL에 대한 분석을 통해 실행 할 수 있는 단계를 만드는데 여기에서 소프트 파싱과 하드 파싱이 존재 합니다. 쉽게 말해서 소프트 파싱은 이미 사용한 내역이 있으면 Shared Pool에서 정보를 재사용 하고 사용한 내역이 없을 경우 하드 파싱을 해서 데이터 파일의 블럭을

Shared Pool로 올려서 구문 분석을 수행 하는 것을 의미 합니다.

## 4.6. LRU 알고리즘

- 최근에 가장 적게 사용한 메모리 공간을 필요 시 가장 먼저 사용하는 오라클 메모리 관리 방법이라고 할 수 있습니다. 기본적으로 LRU 리스트의 양끝을 각각 MRU(Most Recently Used) END, LRU(Least Recently Used) END라고 합니다.

쉽게 설명하자면 동일한 문장이 들어오게 되면 LRU 리스트에서 사용 되었던 메모리 조각을 맨 앞 MRU END로 옮기게 되고 동일한 문장이 아닐 경우 LRU 리스트에서 마지막 메모리 조각을 맨 뒤로 LRU END 이동하고 새로운 조각을 MRU END로 옮기게 됩니다.

## 4.7. ASMM이란 ?

- ASMM (Automatic Shared Memory Management)  
redo log buffer, recycle buffer cache keep buffer cache, flash back buffer 은 자동으로 관리 할 수 없습니다.
- 오라클 10g의 새로운 기능으로 오라클이 업무 부하를 판단하여 shared pool, default data buffer cache, large pool, java pool의 크기를 자동 조정하는 기능을 합니다.
- 관련 프로세스는 MMON, MMAN이 있습니다. 9I R2 부터 자동으로 메모리를 관리 할 수 있었으며, 10G 부터는 ASMM의 새로운 기능이 발달 되었습니다.
  - MMON : 원리 각 메모리에 대한 통계정보를 모으는 것을 의미 합니다. (AWR에 정보를 넣어 줍니다.)
  - MMAN : 브로커들이 각 항목의 데이터를 전달해주면 MMAN이 사이즈를 증가 또는 축소 시켜 줍니다.

## 4.8. Data Buffer Cache 대해서 설명 하세요.

- 모든 사용자가 수행하는 테이블 및 각종 오브젝트들은 이곳에 있어야 하며 만약 이곳에 존재하지 않는다면 서버 프로세스가 하드디스크의 데이터 파일에서 해당 데이터가 들어있는 Block을 찾아서 복사하게 됩니다. 이렇게 운영하는 이유는 수 많은 쿼리를 수행 할 때 메모리를 사용하지 않고 데이터 파일에서 결과를 가지고 오고 넣고 한다면 I/O가 급격히 증가해서 속도가 느려지게 됩니다.
- Data Buffer Cache Block 3가지 상태를 확인 해보겠습니다.
  - Free Buffer는 데이터 파일에 저장되어서 덮어써도 상관없을 경우를 의미 합니다. 만약 Server Process가 데이터 파일에서 Data Buffer Cache로 데이터를 가져오려면 반드시 Free Buffer 상태를 필요한 만큼 확보해야 합니다.
  - Dirty Buffer는 Server Process가 데이터를 변경한 후 데이터 파일로 내려쓰지 않은 상태이기 때문에 절대로 덮어쓸 수 없습니다. 결국에 Free Buffer가 공간이 부족하거나 Dirty Buffer 임계 값을 초과 할 경우 DBWR이 Data Buffer Cache의 Dirty Buffer를 데이터 파일로 내려 쓰게 되는데 다 쓰고 나면 Free Buffer로 변경 됩니다.

- Pinned Buffer는 Server Process가 현재 사용 중인 Block을 의미 합니다.  
순환은 Free Buffer -> Pinned Buffer -> Dirty Buffer 로 사용하게 되며 Dirty Buffer를 데이터 파일로 내려 쓰게 되면 Free Buffer로 변경 되어서 다시 순환을 합니다.  
(서진수 샘 - 오라클 백업과 복구 41P, 52P)

#### 4.9. Redo Log Buffer에 대해서 설명 하세요.

- 복구를 위한 공간이라고 보시면 됩니다. 만약 DML이나 DDL이 발생하게 되면 각 서버 프로세스들이 Redo Log Buffer에 해당 변경 내용을 먼저 기록한 후에 후속작업을 진행 합니다.

#### 4.10. 오라클 Background Process의 종류를 말하세요.

- DBWR, LGWR, CKPT, PMON, SMON이 오라클에서 가장 중요한 역할을 하는 백그라운드 프로세스입니다.
- (서진수 샘 - 오라클 백업과 복구 51-60P )

#### 4.11. 오라클 Background Process의 각각의 역할을 설명 하세요.

- DBWR(Database Writer) 또는 DBWn 이라고 하며, Data Buffer Cache에 있는 저장되어야 하는 내용을 데이터 파일로 저장하는 역할을 하게 됩니다.
  - Checkpoint 신호가 발생했을 때
  - Dirty Buffer 임계값이 지났을 때
  - Time Out이 발생했을 때
  - RAC PING이 발생했을 때
  - Tablespace가 Offline 될 때
  - Tablespace가 Begin backup 상태가 될 때
  - Drop table이나 Truncate table 될 때
- 결국에는 위와 같은 상황이 발생 되면 DBWR 프로세스가 작업을 시작 하게 되는데 중요한 것은 설정에 따라서 여러 개의 프로세스를 사용 할 수 있습니다. DBW1 ~ DBW9를 동시에 사용하여 성능을 향상 시킬 수 있습니다. 관련 Parameter는 db\_writer\_processes를 이용해서 조정이 가능 합니다.
- LGWR(Log Writer)은 Redo Log Buffer의 내용을 Redo Log File로 저장 하게 해줍니다. 그 중에서 어떠한 역할을 할 때 내려쓰는지 알아 보도록 하겠습니다.
  - Commit이 발생 했을 때
  - 1/3이 찼을 때
  - 변경량이 1M가 되었을 때
  - 3초마다
  - DBWR이 내려쓰기 전에

LGWR은 빠른 커밋(Fast Commit), 선 로그 기법(Log Ahead), 동시 쓰기 3가지의 Redo Log Buffer에 서 사용하는 아키텍처를 확인 해볼 수 있습니다. (각각 어떤 역할을 하는지 반드시 숙지 하시기 바

입니다.) ( 오라클 초보자를 위한 10g – 143p ~ 148p)

SCN의 의미와 원리 및 내용은 요약 하자면 commit을 할 때 생성되는 번호를 오라클은 System Commit Number이라고 부르며, 구성은 Base(4Bytes) + Wrap(2Bytes)로 되어있으며 초당 16,000회를 발생시켜도 약 500년 이상 사용할 수 있을 정도입니다. 다른 의미의 SCN으로 System Change Number가 있습니다. datafile, redo log file, controlfile간의 동기화 정보를 맞추기 위해서 사용 됩니다. 구성은 SCN\_BASE(4Bytes) + SCN\_Wrap(2Bytes) + SCN\_Sequence(1Bytes)로 되어있습니다.

- CKPT(Checkpoint Process)는 DBWR에게 checkpoint signal을 발생 시키고 datafile 헤더와 controlfile 헤더에 checkpoint 정보를 기록 해주는 역할을 합니다.
  - 로그 스위치가 발생 할 경우
  - 3초마다 발생
  - 테이블스페이스가 오프라인으로 변경 될 때
  - 데이터베이스가 정상 종료 될 때
  - 사용자가 ALTER SYSTEM checkpoint 명령으로 명시적인 checkpoint를 발생 시킬 때
  - checkpoint 관련 파라미터에서 정한 값에 의해 활동 주기가 되었을 때
- PMON(Process Monitor)는 말 그대로 프로세스들을 모니터링 하고 있다가 문제가 발생하면 해결해 주는 역할을 하게 됩니다.
  - 실패한 프로세스가 수행 하던 트랜잭션 롤백
  - 실패한 프로세스가 획득했던 메모리, 락 및 기타 자원 할당 해제
- SMON(System Monitor)는 오라클 서버가 운영 도중에 비정상적으로 종료되었을 경우(이런 경우를 Instance Cache) 다시 오라클 서버를 startup 시키면서 정리 안되었던 내용을 수정해 주는 역할을 담당합니다.
  - 인스턴스 복구 수행
  - 데이터 Block의 연속된 공간 통합
  - 임시 세그먼트 제거
  - OPTIMAL 크기 유지

SMON의 가장 중요한 역할 중 하나인 인스턴스 복구 작업을 정리 해보도록 하겠습니다. 만약 비정상 종료되었고 다시 DB OPEN을 하면 MOUNT단계에서 OPEN 단계로 넘어 변경 될 때 SMON은 아래와 같은 작업을 진행 합니다.

  1. Redo log file에 있던 모든 내용을 데이터 파일에 전부 적용(Roll Forward)
  2. DB를 오픈
  3. Undo Segment를 확인해서 commit 안 된 데이터는 취소(Roll backward)

## 4.12. 오라클 시작과 종료에 대해서 절차적으로 설명 하세요

- 오라클은 기본적으로 4가지 단계를 의미 합니다. Shutdown, NO Mount, Mount, Open 으로 단계적으로 DB를 Open 시킬 수 있습니다.
- Shutdown은 DB가 종료된 상태입니다. 그래서 cold backup이나 전체 백업 등등을 작업 할 수 있습

니다.

NO Mount 단계는 파라미터 파일 읽기, 인스턴스 생성(SGA, Background Process 가동), alter log와 trace log 추적 및 기록 시작을 합니다.

Mount 단계에서는 Control file를 읽고 datafile과 redo log file의 위치 정보 및 상태를 인지 합니다.

Open 단계는 반드시 datafile, redo log file이 있어야 하며, SMON에 의해서 인스턴스 복구를 자동으로 시작한 후 정상적으로 DB를 Open 합니다. 최종 데이터베이스를 액세스 할 수 있게 됩니다.

(기술면접에서 반드시 나오는 부분이기 때문에 세부적인 항목을 기억 해두시기 바랍니다.)

### 4.13. UNDO Retention은 무엇입니까?

- 오라클 성능고도화 원리와 해법 1 44p 참고, 실전 오라클 백업과 복구 142p 참고
- commit 수행 후에도 해당 undo segment 내에 데이터를 다른 서버프로세스가 덮어쓰지 못하도록 막아주는 시간입니다. 이 부분이 문제가 생기면 오라클은 ORA-0155 Snapshot to old라는 에러가 발생시키게 됩니다.

### 4.14. Redo와 Undo의 차이점

- (성능고도화 1편 36-49P),(서진수 쌤 책 - 112p~114p, 136p~142p)
- redo는 "다시하다"의 뜻으로 풀이 할 수 있고 undo는 "원 상태로 돌리다"의 뜻을 가질 수 있습니다. redo의 가정 기본적인 역할은 복구(recovery) 이며, undo는 작업을 Roll Backward, CR(읽기 일관성), 복구를 합니다. 여기에서 각각의 복구의 의미는 다릅니다. redo는 사용자가 했던 작업을 다시 하지만 undo는 사용자가 했던 작업을 반대로 작업하는 의미로 사용자가 작업을 원상태로 돌리게 됩니다.
- redo는 기본적으로 복구의 역할을 하며 DML 작업을 하게 되면 리두에 기록 하게 됩니다.
- undo는 작업 롤백, 읽기 일관성, 복구를 합니다.
- <http://hactivist.tistory.com/tag/redo> (참고)

### 4.15. Index 사용이유?

- 간단히 말해서 인덱스의 가장 큰 장점은 수 많은 데이터 중에서 원하는 데이터를 빨리 액세스 할 수 있게 해주는 것이 인덱스 입니다. 예를 들면 책을 볼 때 필요한 내용만 보기 위해서 목차만 보고 몇 페이지에 정보가 있는지 아는 것 처럼 인덱스도 그러한 장점을 가지고 있습니다. 또한 인덱스는 사원 테이블에서 사원 컬럼(인덱스)을 순서대로 정렬해서 사원 테이블의 각 로우를 바로 액세스 할 수 있는 주소를 ROWID 정보와 함께 저장 하게 됩니다.

### 4.16. Index의 종류와 그 기능

- ROWID : 해당 로우를 액세스할 수 있는 주소 값
  - 오프젝트 번호, 파일번호, Block번호, 로우번호 구조로 되어있습니다.

- B\*트리 인덱스
  - 구조는 크게 루트(root) Block, 브랜치(Branch) Block, 리프(Leaf) Block으로 되어있습니다.
  - 온라인 업무에서와 같은 적은 로우 데이터 액세스 시 유리
  - 분포도가 나쁜 컬럼에 대해서는 성능 저하 발생 가능
  - 하나의 로우 액세스시 어느 로우나 동일한 양의 Block 액세스
  - 종합적으로 얘기 하자면 B\*트리 인덱스는 적은 범위를 액세스할 경우에 빠른 조회 성능을 보장하지만 반대로 성별과 같은 데이터의 종류가 적고 중복성이 큰 컬럼에는 성능 저하가 발생합니다.
- 비트맵 인덱스
  - 구조는 루트, 브랜치, 리프 Block과 같이 B\*트리 인덱스와 동일 합니다.
  - 분포도가 나쁜 컬럼에 대해서 성능 보장
  - CR 연산에 대해서 효율적
  - DML 작업에 의한 데이터 변경 시 부하 급증 가능
  - 분포도가 나쁜 컬럼에 대해서 유리한 구조를 가지며 비트 연산 방식을 사용하기 때문에 OR 연산에 유리 합니다. 하지만 DML 발생시 모든 비트맵에 변경이 발생하여 성능에 저하를 발생 시킬 수도 있습니다.
- 역 전환 키 인덱스
  - 쉽게 말해서 사원 번호 인덱스의 1234번 사원번호는 실제 테이블에서는 4321로 되어 있다고 생각 하면 됩니다.
  - 좌측 리프 Block에 대한 경합 해소(B\*트리 인덱스에 유리)
  - 범위 스캔시 인덱스 액세스 불가
- 함수 기반 인덱스
  - sql문의 where 조건절에서 좌측에 비교 대상 컬럼을 변경하게 되면 해당 컬럼에 인덱스가 존재하여도 인덱스를 사용하지 못하고 테이블 전체를 스캔하게 됩니다. 인덱스에는 변경된 값이 아닌 원래의 컬럼 값이 저장고 변경된 값으로는 비교할 수 없기 때문입니다.
  - 함수 기반 인덱스가 장점은 계산 값에 대한 인덱스 스캔이 가능 한 것과 논리적인 모델링의 문제를 극복 할 수 있으며, 단점은 DML 부하가 발생하고 인덱스의 유연성이 저하가 됩니다.

## 4.17. Index SPLIT 대한 정의 및 발생원인, 대해 설명 하세요

- index split은 인덱스를 분할 한다라는 의미를 가지고 있습니다. 하나의 리퍼 Block을 2개로 분할 합니다.
- 하나의 리퍼 Block에 3개의 키만 입력 가능하다고 가정하고 첫 번째 키를 insert 하고 두 번째, 세 번째 키를 모두 insert 했습니다. 그럼 리퍼 Block에는 3개의 키가 저장 되어서 Block이 꽉 차게 되었습니다. 이 상태에서 4번째 키를 insert 하면 새로운 Block을 확보 합니다. 이후 2개의 리퍼 Block에 저장하게 됩니다. 새로운 브랜치 Block이 추가가 되는 것을 알 수 있습니다. 결국에는 인덱스 split이란 새로운 인덱스가 자란다는 의미와 같습니다.

- 인덱스 split은 무거운 작업이기 때문에 오라클은 split을 줄이려고 하고 있습니다. 랜덤하게 추가되면 50:50으로 하고 최대값이 추가 될 경우에는 90:10으로 합니다. 오라클은 2가지 방법으로 하는데 50:50은 반반으로 대충 맞게 한다는 의미고 90:10은 오른쪽으로 쏠리는 현상이 발생 됩니다.
- 하지만 이러한 문제를 해결 할 수 있는 방법은 우리가 할 수 없습니다. 인덱스 split의 경합을 줄이기 위해서는 동시에 split을 하는 session의 수를 줄이거나 불필요한 인덱스를 삭제 할 수 있고 인덱스 생성할 때 pctfree를 높게 부여 하면 해결 할 수 있습니다. pctfree를 높게 준다는 의미는 만약 pctfree의 값을 90으로 준다면 인덱스 리퍼 Block의 공간은 10을 제외한 나머지는 텅텅 비어 있을 겁니다. 따라서 split가 발생하면 공간을 사용하기 때문에 줄일 수 있습니다.
- 가장 중요한 방법은 rebuild를 해주는 것이 좋습니다. 하지만 빌드를 하면 크기를 늘어나게 됩니다.  
<http://www.youtube.com/user/ukjax>

#### 4.18. FRA 란

- Flash Recovery Area 오라클 10g 부터 나오기 시작 하였고 데이터베이스 복구와 백업 관련 파일을 저장하는 영역 입니다.
- archived log files을 flash recovery area에 생성하려면 2가지 방법이 있습니다.
  - log\_archive\_dest\_1 = 'LOCATION=/app/oracle/oradata/arch'
  - log\_archive\_dest\_n 파라미터를 아예 설정하지 않으면 내부적으로는 db\_recovery\_file\_dest의 값으로 설정한다고 합니다. 포맷 또한 영향을 받지 않고 일별로 폴더가 생기면서 OMF 파일형식으로 생성 됩니다.

#### 4.19. ADDM이란?

- Automatic Database Diagnostic Monitor는 각 AWR 스냅샷 생성 후마다 실행 되며 instance를 모니터링하여 병목지점을 감지 합니다.
- 결국에 AWR에 결과를 저장 합니다. (EM으로 볼 수 있음)
- <http://kr.forums.oracle.com/forums/thread.jspa?threadID=477125&tstart=105>

#### 4.20. Oracle 리오그(Reorganization) 작업은?

- 리오그 작업은 테이블 기준으로 보면 exp/imp, table move, CTAS(CREATE TABLE AS SELECT \* FROM) 3가지로 나눌 수 있습니다.
- 여기에서 중요한 것은 리오그 작업은 가능하면 하지 않는 것을 권고 하고 있습니다. 보편적으로 [클러스터링 팩터\(인덱스와 테이블간에 순서가 비슷하면 좋음\)](#) 낮출 때 효과가 있습니다.
- 인덱스 rebuild에 대해서는 여러 가지 생각을 해야 합니다. 리빌드를 하게 되면 리퍼 Block들이 팍 차게 되서 다시 DML에 의해 split이 자주 일어나면 성능이 저하될 수도 있습니다. 그래서 인덱스 리빌드는 많은 case를 생각해볼 만한 가치가 있다고 생각 합니다.
- <http://outspace.egloos.com/2255045>



## 4.21. Select 의 문장이 실행되는 과정?

- SELECT name FROM DEPTNO WHERE B='park'
- Parse -> Bind -> Execute -> Fetch
- User Process는 사용자가 실행한 SQL을 가지고 DB 서버에 접속하여 리스너를 통해서 Server Process에게 쿼리를 전달 합니다. 이후 Server Process는 각 단계별로 작업을 시작 합니다. 우선 Shared Pool에서 구문분석, 문법검사, 권한검사, 커서공유, 실행계획을 진행 합니다. 이러한 단계를 오라클에서는 Parse라고 합니다. Bind는 값을 치환하여 변수 값을 적용 하며 만약 바인드 처리를 하지 않았다면 바로 실행 단계로 넘어 갑니다. Execute는 Data Buffer Cache에서 데이터파일에 필요한 Block을 적재하거나 만약 데이터가 Data Buffer Cache에 있다면 재사용 하게 됩니다. 최종적으로 Fetch는 필요한 데이터만 골라서 User Process에서 전달합니다.
- 실전 오라클 백업과 복구 20P ~ 24P 참조
- <http://develop.sunshiny.co.kr/575>

## 4.22. Update(DML)의 문장이 실행되는 과정

- UPDATE DEPTNO SET name='sang' WHERE B='park'
- Parse -> Bind -> Execute
- User Process는 사용자가 실행한 SQL을 가지고 DB 서버에 접속하여 리스너를 통해서 Server Process에게 쿼리를 전달 합니다. 이후 Server Process는 각 단계별로 작업을 시작 합니다. 우선 Shared Pool에서 구문분석, 문법검사, 권한검사, 커서공유, 실행계획을 진행 합니다. 이러한 단계를 오라클에서는 Parse라고 합니다. Bind는 값을 치환하여 변수 값을 적용 하며 만약 바인드 처리를 하지 않았다면 바로 실행 단계로 넘어 갑니다. Execute 단계는 아래에서 설명 하겠습니다.
- Select 문장과 달리 fetch 단계가 없고 Execute 단계에서 종료가 됩니다. 하지만 Execute 단계는 Select 문장과 조금 다르게 동작 합니다. 사용자가 update 쿼리를 실행하며 서버 프로세스가 우선 가장 먼저 사용 가능한 undo segment부터 확보하게 됩니다. 변경을 된 내용을 리두 로그 버퍼에 기록하고 이전 데이터를 undo segment에 기록 합니다. 마지막으로 DB Buffer Cache에 변경된 내용을 기록 하게 됩니다.
- 실전 오라클 백업과 복구 25p ~27p

## 4.23. Commit 실행 되는 과정

- 보통 commit를 수행하면 데이터 파일로 저장한다고 생각 하는 분들이 많지만 정확한 표현은 commit를 수행하면 리두 로그 버퍼에 존재하는 변경에 대한 로그를 리두 로그 파일에 기록하게 됩니다. 이는 빠른 커밋의 아키텍처와 지연 쓰기에 의해 실제 데이터 버퍼 캐쉬에 존재하는 변경된 데이터를 디스크에 저장하지는 않습니다.

## 4.24. 오라클의 커서는 무엇인가?

- 오라클에서 말하는 커서는 공유커서(shared Cursor), 세션커서(Session Cursor), 애플리케이션커서(Application Cursor) 로 나뉠 수 있습니다.
- 공유커서란 라이브러리 캐시에 공유돼 있는 Shared SQL Area
- 세션커서란 Private SQL Area에 저장된 커서
- 애플리케이션커서란 세션 커서를 가르키는 핸들
- 오라클 성능 고도화 원리와 해법 1 257 ~ 271 (꼭 읽어 보세요.)

## 4.25. 오라클 트리거란?

- 트리거는 insert, update, delete문이 table에 대해 행해질 때 묵시적으로 수행되는 Procedure 입니다.
- 주로 사용되는 용도에 대해서 알아 보겠습니다.
  - 로그성 테이블을 만들 때(insert, update, delete 해당 로우에 대한 로그를 남길 때)
  - 집계성 테이블을 만들 때(일자별, 실적에 트리거 달아서 월별 실적 테이블에 데이터를 만들 때)
  - 트랜잭션에 대한 데이터 무결성이나 동기화가 필요할 때(트랜잭션이 A-B-C-D 테이블을 거쳐 갈때 )
  - 테이블 까지 모두 데이터가 동기화 되어야 하거나 무결성이 반드시 보장 되어야 할 때, 주로 리얼 타임 온라인 업무
  - 오라클 트리거는 좋긴 하지만 너무 많이 쓰면 시스템이 전체적으로 느려지는 경우가 발생 하기 때문에 실제로 사용 할 때는 트리거를 튜닝해야 합니다.
  - <http://skkim.tistory.com/entry/%ED%8A%B8%EB%A6%AC%EA%B1%B0-%EC%98%A4%EB%9D%BC%ED%81%B4>

## 4.26. 오라클 동의어(Synonym)란?

- 다른 유저의 테이블을 조회할 경우 테이블소유자.테이블이름으로 조회하는 것을 자신의 테이블과 같이 조회할 수 있도록 만들어 주는 데이터베이스 오브젝트입니다.

## 4.27. 오라클 시퀀스(Sequence)란?

- 시퀀스는 주로 사원번호, 일련번호 등등 중복되지 않는 유일한 값을 가지기 위해서 사용하는 경우가 많습니다. 보통 옵션에 따라서 순차적으로 번호를 할당하는 오라클 오브젝트의 한 종류라고 생각 하시면 됩니다. 여기서 중요한 것은 NEXTVAL, CURRVAL의 차이점을 알아야 합니다.
- NEXTVAL : 사용 가능한 다음 시퀀스 번호를 할당 받기 때문에 연속적인 시퀀스 번호를 추출하는데 사용
- CURRVAL : 현재 세션에 할당된 시퀀스 번호를 추출하므로 NEXTVAL을 호출한 후 사용 가능

- NEXTVAL, CURRVAL의 사용 할 수 있는 SQL문은 아래와 같습니다.
  - 서브 쿼리가 아닌 SELECT문의 SELECT List에 사용
  - INSERT절에 있는 서브쿼리의 SELECT List에 사용
  - INSERT절의 Values절에 사용
  - UPDATE절의 SET절에 사용
- NEXTVAL, CURRVAL이 사용 할 수 없는 SQL문은 아래와 같습니다.
  - 뷰의 SELECT List
  - DISTINCT가 있는 SELECT문
  - GROUP BY 또는 ORDER BY가 있는 SELECT문
  - 서브쿼리

#### 4.28. 익스텐트 관리 방법의 2가지를 설명하세요.

- DMT(Dictionary Management Tablespace), LMT(Local Management Tablespace)
- DMT : 사용 가능한 또는 할당된 익스텐트의 정보를 시스템 테이블스페이스에 저장된 데이터 디렉터리 테이블에서 관리 하는 방식 입니다.
- LMT : 지역 관리 테이블스페이스에 존재하는 모든 세그먼트는 SYSTEM 테이블스페이스의 데이터 디렉터리 테이블에 익스텐트의 정보를 기록하지 않아도 된다. 테이블스페이스의 데이터 파일 헤더에 존재하는 비트맵에 해당 익스텐트의 사용 유무를 기록하게 됩니다.
- 쉽게 말해서 DMT는 학원에서 수강하는 모든 정보를 한 곳에서 관리하기 때문에 강의실, 강의실 번호, 담당 선생님 등등을 알고 있어야 한다면 LMT는 학원에서 안내원이 강의실만 가르쳐 주면 강의실 앞에 자리 배치도를 통해서 수많은 학생이 들어와도 쉽게 처리 할 수 있다는 장점이 있습니다.
- 초보자를 위한 오라클 10g 335p ~ 338p 참조

#### 4.29. 세그먼트 공간 관리 방법의 2가지를 설명하세요.

- 세그먼트 공간 관리 방법은 FLM과 ASSM으로 구분 할 수 있습니다.
- FLM(FREELIST Management)는 세그먼트를 구성하는 익스텐트 중 첫 번째 익스텐트의 첫 번째 데이터 Block인 세그먼트 헤더에서 프리리스트로 여유 공간을 가지는 데이터 Block을 관리하는 방식 입니다. (9i에서 주로 사용)
- ASSM(Automatic Segment Space Management)는 세그먼트를 구성하는 각 익스텐트의 첫 번째 데이터 Block인 익스텐트 헤더에서 비트맵으로 여유 공간을 가지는 데이터 Block을 관리하는 방식 입니다. (10g 부터 주로 사용)
- <http://www.gurubee.net/pages/viewpage.action?pageId=6259679>

#### 4.30. 정렬 절차에 대해서 설명하세요.?

- 일반적으로 정렬작업과 imp/exp 등 작업을 하기 위해서는 PGA 영역의 sort에서 수행하지만 많은 데이터를 처리 할 수 없기 때문에 임시 테이블스페이스를 사용하게 됩니다. 따라서 임시 테이블스

페이지의 종류를 확인 하겠습니다.

- 기본 임시 테이블스페이스 : 데이터베이스 생성시 지정된 기본 임시 테이블스페이스
- 로컬 임시 테이블스페이스 : 유저를 생성시 지정하는 임시 테이블스페이스
- 기본 임시 테이블스페이스는 많은 트랜잭션에서 사용하기 때문에 유저를 생성하게 되면 반드시 로컬 임시 테이블스페이스를 생성하는 것이 바람직합니다. 임시 테이블스페이스와 달리 로컬 임시 테이블스페이스는 Data Buffer Cache를 거치지 않고 다이렉트로 액세스를 한다는 장점이 있습니다.
- 정렬 절차를 설명 하겠습니다.
  - 인덱스를 생성했다고 가정 하겠습니다. CREATE INDEX idx\_dept ON DEPT(name) TABLESPACE ts1
  - 테이블에 액세스를 하는 순간 PGA 정렬 공간에서 인덱스 생성을 위한 정렬을 시작 합니다. PGA의 공간이 부족하게 되면 임시 테이블스페이스로 서버 프로세스가 정렬된 Block을 기록 합니다.
  - PGA의 공간을 임시 테이블스페이스로 내려쓰는 절차를 계속 하게 되고 모두 정렬이 되었다면 머지 소트 방식을 이용해서 임시 테이블스페이스의 전체 데이터를 정렬하게 됩니다.
  - 모든 정렬이 종료되면 서버 프로세스에게 전달하게 됩니다.
- 초보자를 위한 오라클 10G 349P~351P

#### 4.31. 래치((Latch)와 락(Lock)

- 래치 : 오라클 Concept 매뉴얼에서는 래치를 가벼운 락(Lightweight lock)으로 분류하고 있습니다. 즉 락(Lock) 중에서도 아주 빠른 속도로 작동하게끔 구현된 경량화된 락이라는 것입니다. 래치는 물리적으로 Shared Pool 영역에 존재하는 일종의 메모리 구조체입니다. 래치는 매우 간단하고 작은 메모리 영역을 사용하며, 래치를 획득(get)하고 해제(release)하는 작업들 또한 하드웨어에 맞게 극도로 최적화되어 있습니다.
- 락 : 락은 두가지로 구분할 수 있는데 첫째, 락에 대한 정확한 분류가 존재하지 않는다는 것과 둘째, Enqueue 락과 일반 락과의 정확한 구분이 존재하지 않는다는 것에 기인합니다. 락은 다시 말해서 래치(Latch)가 보호하는 리소스(resource) 이외의 모든 데이터베이스 오브젝트를 의미 합니다. 테이블 스페이스, 테이블, 로우, 트랜잭션, LCO(Library Cache Object), RCO(Row Cache Object)로 구분 할 수 있습니다.

#### 4.32. Incremental Checkpoint와 Normal Checkpoint의 차이점?

- 체크포인트는 다음과 같은 두 종류로 구분 할 수 있습니다.
  - Incremental Checkpoint : 매 3초마다, 로그스위치 발생시
  - Normal Checkpoint : 데이터베이스 정상 종료, ALTER SYSTEM CHECKPOINT; 명령어 실시
- Incremental Checkpoint는 데이터베이스가 정상 운영 되고 있는 상태에서 주기적으로 발생하며, 동작원리는 다음과 같습니다.
  - 매 3초가 되면, CKPT 백그라운드 프로세스는 DBWR에게 작업 신호를 전송합니다.

- DBWR는 데이터베이스가 현재 상태에서 비정상 종료되었을 경우, FAST\_START\_MTTR\_TARGET 파라미터에 정의된 시간(초)이내에 인스턴스 복구가 가능한지 확인해서, 불가능한 경우 데이터베이스 버퍼 캐시의 변경된 Block 일부를 데이터 파일에 기록하게 됩니다. 만약 파라미터에 정의된 시간 내에 복구가 가능한 경우, DBWR는 작업 신호를 무시하게 되며, CKPT는 컨트롤 파일과 데이터 파일 헤더를 갱신하지 않습니다. 단, 로그스위치에 의해 발생한 체크포인트인 경우, LSN(Log Sequence Number), SCN(System Change Number) 를 컨트롤파일과 데이터 파일 헤더에 기록하게 됩니다.
  - DBWR 백그라운드 프로세스가 동작하여 가장 오래 전에 변경된 Block의 일부(인스턴스 복구가 FAST\_START\_MTTR\_TARGET 파라미터에 정의된 시간에 완료 될 수 있을 만큼만 데이터베이스 버퍼 캐시 내의 변경된 Block의 일부를 데이터 파일에 기록)를 데이터 파일에 기록하였다면, CKPT 백그라운드 프로세스는 컨트롤 파일의 체크포인트 위치를 갱신하게 됩니다. 체크포인트, 위치란 리두 로그 파일 내에서 데이터 파일의 마지막 변경 부분에 해당하는 부분을 가리키는 위치입니다. 즉, 데이터베이스가 비정상 종료되었을 때, SMON 백그라운드 프로세스에 의해 인스턴스 복구가 시작되는 위치입니다.
  - 실제 체크포인트가 발생하는 간격은 3초가 절대 아니고, FAST\_START\_MTTR\_TARGET 파라미터에 따라 발생 주기는 달라지게 되는 것입니다.
- Normal Checkpoint는 일반적으로 데이터베이스를 정상 종료하는 경우 발생하며, 동작원리는 다음과 같습니다.
- ABORT 옵션을 제외한 SHUTDOWN 명령으로 데이터베이스를 종료 합니다.
  - 리두 로그 버퍼의 모든 변경 사항을 LGWR 백그라운드 프로세스가 데이터 파일에 모드 기록 합니다.(이 부분이 조금 이해가 되지 않아서 나름대로 해석 해보겠습니다. 리두 로그 버퍼의 모든 변경 사항을 LGWR 백그라운드 프로세스가 로그 파일로 내려 쓰게 됩니다.)
  - 데이터베이스 버퍼 캐시의 모든 변경된 Block을 DBWR 백그라운드 프로세스가 데이터 파일에 모드 기록 합니다.
  - CKPT 백그라운드 프로세스가 컨트롤 파일의 체크포인트 위치를 리두 로그 파일의 가장 마지막 부분으로 갱신하고, SCN LSN을 컨트롤 파일 및 데이터 파일 헤더에 기록 합니다.
- 결과적으로 FAST\_START\_MTTR\_TARGET 파라미터의 값을 증가시키면 체크포인트 간격은 증가되고, 데이터베이스의 성능은 향상되지만 데이터베이스가 비정상 종료되었을 경우, 인스턴스 복구에 소요되는 시간이 길어지게 됩니다, 반면 파라미터의 값을 감소시키면 체크포인트 간격은 감소되고, 데이터베이스의 성능은 저하되지만 데이터베이스가 비정상 종료되었을 때, 인스턴스 복구에 소요되는 시간은 짧아지게 되는 장점이 있습니다.
- <http://cafe.daum.net/ocp/2rYV/490> 김영조 샘

## 5. 오라클 백업과 복구

### 5.1. Archive log 모드와 No Archive log 차이점

- 오라클에서 복구의 핵심은 redo log 입니다. 하지만 redo log의 단점은 로그 스위치가 계속 일어나면 해당 그룹의 내용을 재사용 하기 때문에 오랜 시간 기록을 할 수 없습니다. 그래서 나온 것이 Archive Log 모드 입니다. 로그 스위치가 발생하면 redo log 파일의 내용을 archive log 파일로 내려쓰게 됩니다. no archive log mode는 그래서 redo log file을 덮어쓰는 경우를 의미 합니다. 어떠한 백업 파일도 없이 계속 덮어쓰게 되는 것입니다. 복구에서 가장 중요하다고 할 수 있는 것이 순차적으로 한다는 것인데요. 중간에 로그 파일이 비어 버리게 되면 빈 파일 다음부터는 복구를 할 수 없습니다. 그래서 archive log mode를 이용하면 로그 파일을 덮어 쓰기 전에 다른 곳으로 복사를 해주기 때문에 복구가 가능 한 것입니다.
- archive log mode의 단점은 archive log file를 저장해야 하는 별도의 저장 공간이 필요하다는 것과 archive hang이 발생할 수 있습니다. 마지막으로 관리가 까다롭기 때문에 영구적으로 복구를 수행하지 못할 수도 있습니다. 단점보다 큰 장점이 많기 때문에 archive log mode를 이용하는 것을 권장합니다.

### 5.2. Archive Hang 원인과 해결 방법은?

- Redo log 의 기본 구조를 먼저 설명하겠습니다. Redo log는 1그룹 당 여러 개의 멤버를 생성 할 수 있습니다. 그렇다면 여러 개의 그룹에 있는 여러 개의 파일에 있는 내용은 모두 동일하게 기록 됩니다. 보통 오라클에서는 2개이상의 그룹이 생성되어 있어야 하기 때문에 여기에서는 3개의 그룹이 있다고 가정하고 말씀 드리겠습니다. commit 트랜잭션이 발생하여 Redo Log Buffer에 있는 내용을 LGWR이 내려쓰게 됩니다. 그룹 1번이 용량이 꽉 차서 다른 그룹으로 넘기는 것을 로그 스위치라고 말할 하게 됩니다. 그룹 2번으로 넘겨 졌다면 그룹 1은 ACTIVE이고 그룹 2번은 CURRENT상태가 될 것이고 그룹 3번은 INACTIVE 상태가 됩니다. 이렇게 로드 밸런싱에 의해서 redo log buffer의 데이터를 기록 하게 됩니다.
  - 그룹 1 : ACTIVE (현재 Archive log file 저장하는 공간이 가득 차있음)
  - 그룹 2 : ACTIVE(그룹 1에서 로그 스위치에 의해 넘어왔지만 역시 공간이 없음)
  - 그룹 3 : CURRENT(현재 파일에 내려쓰는 중)
  - 이러한 상황에서 로그 스위치가 일어나면 다른 그룹으로 내려 쓰려 하려고 하지만 다른 그룹도 현재 Archive log file에 내려쓰지 못하는 상황이기 때문에 행(Hang)이 발생 하는 것입니다.
- 해결방법
  - Archive log file을 백업 받고 Archive log file를 삭제 합니다. 이후 sqlplus internal 로그인 하여 "alter system archive log start" 를 수행 후 reference를 합니다.
  - 이 방법은 아카이브 로그 파일이 저장되지 않게 하는 방법입니다. alter system set LOG\_ARCHIVE\_DEST\_STATE\_1 = DEFFER; 이 후 공간을 확보 후 alter system set

log\_archive\_dest\_state\_1 = enable;로 이전처럼 아카이브 로그파일을 사용하게 합니다.

- 아카이브 파일을 다중화한다면 아카이브의 빠른 수행을 위해 하나의 ARCH 백그라운드 프로세스가 하나의 아카이브 로그파일에 저장하는 것이 성능에 유리 하기 때문에 alter system set log\_archive\_max\_processes = 4를 주고 4개의 ARCH 백그라운드 프로세스를 가동하는 방법도 있습니다.

### 5.3. Oracle 닫힌 백업(cold backup), 열린 백업(hot backup)은 무엇인

#### 가요?

- cold backup은 DB를 종료 후 OS명령어로 다른 저장장치로 옮기는 작업을 의미 하고 Hot backup은 DB를 종료 하지 않고 테이블스페이스와 컨트롤파일을 백업 할 수 있는 방법입니다.
- Hot Backup은 원리가 중요합니다. 원칙 1 Archive log 모드여야 한다. 2. 모든 데이터파일을 백업 할 수 없다 3. tablespaces offline와 반대로 정상적으로 작업을 수행 합니다.

3번을 좀 더 얘기 해보도록 하겠습니다. begin backup 후 상태를 보면 checkpoint cnt 값이 변경 되고 checkpoint가 발생하면 db buffer cache에 있는 해당 테이블스페이스 정보가 데이터 파일로 내려 써지게 됩니다. 작업이 완료 되서 최종 SCN 정보가 데이터 파일과 컨트롤 파일에 기록되며 해당 테이블스페이스의 데이터 파일 헤더에 Hot-backup-in-process 라는 플래그 값이 표시 됩니다. 중요한 것은 아직도 begin backup 상태라고 가정하고 최종 SCN 이후로 변경이 된다면 전부 Redo log buffer에 기록 하게 됩니다. 그렇다면 그 많은 데이터를 redo log buffer에 담을 수 없기 때문에 redo log file에 기록하게 되고 로그 스위치가 발생하면 archive log file에 기록 하게 됩니다. 그래서 begin backup은 archive log mode에서 밖에 할 수 없다는 것입니다. 마지막으로 end backup을 수행하면 모두 데이터 파일에 반영되게 됩니다.

### 5.4. 데이터 파일 복구

- no archive log mode 일 경우
  - no archive log 모드일 경우에는 두 가지 방법 밖에 없습니다. 첫 번째는 전체 백업된 시점으로 돌리는 방법과 하나의 데이터 파일만 손실이 발생 했다면 손실된 데이터 파일만 drop 하면 됩니다. 두 번째 방법에서 drop의미는 아래와 같은 명령어를 이용하면 됩니다.
  - alter database datafile 'file\_name' offline drop
  - 최악의 경우에는 오라클 비공식 방법인 Hidden Parameter를 이용해서 올리면 됩니다. pfile에 \_allow\_resetlogs\_corruption=true 추가하고 recover databae until cancel; 을 이용해서 올리면 됩니다.

- archive log mode 일 경우
  - 수 많은 경우의 수가 있으므로 표로 작성 하겠습니다.

D : Datafile, C : Controlfile, R : Redo Log File

D	C	R	해결 방법
<b>완전 복구를 수행 하겠습니다.</b>			
○	○	○	[상황] DB에서 데이터파일(test01.dbf) 삭제 발견 하여 DB OPEN 상태에서 복구를 해야 합니다. [준비사항] 백업 파일 존재 [복구시작] rm -rf로 삭제된 데이터 파일로 인해 에러가 발생하였다면 test테이블 스페이스를 offline 시킨 후 백업 되어있는 데이터 파일을 현재 시스템으로 옮긴 후 recover tablespace test 명령어를 수행하면 redo 로그에 남아있는 데이터를 데이터 파일에 그 내용을 복구 할 수 있습니다.
○	○	○	[상황] SYSTEM 테이블스페이스를 관리자가 삭제 해서 복구를 해야 합니다. [준비사항] 백업 파일 존재 [복구시작] 역시 rm -rf 명령어로 삭제 된 경우라면 역시 system 테이블스페이스를 offline 하면 되지만 undo와 system은 offline이 되지 않기 때문에 DB를 재 가동 합니다. 그럼 MOUNT 단계에서 에러가 발생 하게 되므로 백업 되어있던 system 데이터 파일을 옮긴 후 recover database; 한 후 정상적으로 오픈 시키면 됩니다.
X	○	○	[상황] 일반 테이블스페이스 test의 데이터 파일을 관리자가 삭제 해서 복구를 해야 합니다. [준비사항] 백업 파일 없음 [복구시작] rm -rf 명령어로 삭제 된 경우라면 우선 offline 시킵니다. 불행하다면 백업 데이터파일이 없기 때문에 신규로 테이블스페이스를 생성 후 복구 해야 합니다. alter database create datafile 'file_name' as 'file_name' 명령어를 수행하면 해당 빈 데이터 파일이 생성 됩니다. 이 상태에서 recover tablespace test 하면 redo log에 있는 데이터를 복구하기 시작 합니다.
X	○	○	[상황] 데이터 파일 : /data, 리두 로그 파일 : /redo, 컨트롤 파일 : /cont 중에서 데이터 파일의 disk가 장애가 발생 되었습니다. 복구를 해야 합니다. [준비사항] 백업 파일 존재(단, 신규 TEST 테이블 스페이스는 없는 상황) [복구시작] 우선 모두 /temp 파일로 옮기겠습니다. /redo와 /con는 현재 사용하고 있는 파일이고 /data는 사용할 수 없으므로 백업 되어있던



		<p>/backup 데이터 파일을 /tmp로 옮겼습니다. 이렇게 해서 DB를 open을 하면 MOUNT 단계에서 에러가 발생 합니다. 이유는 컨트롤 파일의 헤더 정보에는 데이터 파일이 장애가 발생한 /data로 되어 있기 때문에 파일을 사용 할 수 없어서 그렇습니다. 그래서 관련 정보를 모두 /temp로 옮기면 됩니다. alter database rename file 'prev_file' to 'next_file' 명령어를 이용해서 모두 /temp로 변경합니다. 하지만 여기에서 문제는 신규 TEST 테이블스페이스는 백업 파일이 없으므로 신규로 빈 파일을 생성합니다.</p> <pre>alter database create datafile 'priv_file' as 'next_file'</pre> <p>이 후 복구를 시작 합니다. recover database; 정상적으로 DB를 Open하면 모두 복구가 완료 되어있을 겁니다.</p> <p>결론을 말씀 드리자면 완전 복구라는 의미는 모든 데이터파일의 내용이 정상적으로 복구 되는 것을 의미 합니다. 시간 기반이나 이전으로 돌아가는 방법은 불완전 복구에 속합니다.</p>
<p><b>여기에서부터는 불완전 복구를 해보겠습니다.</b></p>		
<p>○</p>	<p>○</p>	<p>○</p> <p>[상황] 현재 상황은 사용하고 있는 테이블을 삭제(Drop table)하였고 반드시 복구를 해야 하는 상황입니다.</p> <p>[준비사항] 백업 파일 존재</p> <p>[복구시작] 테이블을 살리기 위해서는 불완전 복구 밖에 없습니다. 단, CLONE DB를 구성 한다면 CTAS(DB LINK)를 이용해서 완전 복구를 할 수도 있습니다. 하지만 여기에서는 현재 상황에서 할 수 있는 방법을 구상해보도록 하겠습니다. 시간기반으로 복구를 할 예정인데 운이 좋게도 로그 마이너를 통해서 DROP된 시간을 확인 할 수 있었다고 가정하겠습니다.(alert 로그에는 drop tablespace 정보만 확인 할 수 있습니다.)</p> <p><b>현재 사용하고 있는 리두 로그 파일</b></p> <p><b>현재 사용하고 있는 컨트롤 파일</b></p> <p><b>백업 된 데이터 파일</b></p> <p>MOUNT 단계에서 복구를 시작 하겠습니다.</p> <pre>recover database until time '2010-10-03:04:43:22';</pre> <p>auto [ Enter 합니다. ]</p> <p>여기에서 auto의 의미는 archive log file를 모두 자동으로 적용해줍니다.</p> <p>cancel은 복구 작업을 중단</p> <p>filename(redo log file, archive log file)을 직접 입력</p> <p><b>alter database open resetlogs (반드시 resetlogs로 올려야 합니다.)</b></p> <p>resetlogs 옵션을 사용하는 가장 큰 이유는 데이터 파일과 리두 로그 파일, 컨트롤 파일에 checkpoint scn 정보를 동일하고 만들어 줍니다. 그 외에 redo log file의 LSN 정보를 0으로 초기화 하며, 디크너리의 데이터 파</p>

			<p>일 정보와 컨트롤 파일의 데이터 파일의 정보를 비교해서 디렉터리에 있지만 컨트롤 파일에 없는 파일은 MISSINGXXXXX으로 가짜 엔트리를 생성합니다.</p>
X	X	O	<p>[상황] Drop Tablespace로 잘못 삭제된 테이블 스페이스를 복구하는 해야 합니다.</p> <ol style="list-style-type: none"> <li>1. 전체 백업 (모두 백업 되어있음)</li> <li>2. DB OPEN 후 신규 테이블 스페이스 생성 후 삭제</li> <li>3. 잘못 지운 신규 테이블 스페이스를 복구</li> </ol> <p>[준비사항] 백업 파일 없음, 컨트롤 파일 백업 없음(전체 백업에는 신규 테이블 스페이스에 대한 정보는 없음)</p> <p>[복구시작] 중요한 사항입니다.</p> <p><b>컨트롤 파일 전체 백업 시점</b></p> <p><b>데이터 파일 전체 백업 시점</b></p> <p><b>리두 로그 파일 현재 시점</b></p> <ul style="list-style-type: none"> <li>● 컨트롤 파일은 왜 현재 시점에 있는 파일을 사용하지 않는지에 대한 이해는 이렇게 생각 하시면 됩니다. 이미 컨트롤 파일에는 DROP TABLESPACE 신규테이블스페이스 명령어를 수행 하였기 때문에 비정상적으로 처리 된 것이 아닙니다. 그래서 현재 시점에 있는 컨트롤 파일로 복구를 한다면 절대로 이전 시점으로 돌아갈 수 없습니다.</li> </ul> <p>이전 전체 백업 시점인 컨트롤 파일을 기준으로 MOUNT단계까지 올리면 신규 테이블 스페이스에 대한 정보는 없습니다. 그래서 아래와 같은 명령어를 이용해서 리두 로그 파일에 있는 정보를 컨트롤 파일에 기록 합니다.</p> <pre>alter database until time '2010-10-10:23:10:10' using backup controlfile</pre> <p>- 상황에 따라서 redo log file를 직접 입력 하거나 auto를 입력 해야 합니다.</p> <p>하지만 백업 데이터 파일에는 신규 테이블스페이스가 없기 때문에 아래와 같은 형태로 controlfile에 추가 될 것입니다.</p> <pre>/home/oracle/oradata/testdb/UNNAMED0007</pre> <p>앞서 한 것 처럼 신규 테이블 스페이스를 생성 하시면 됩니다.</p> <pre>alter database create datafile 'file_name' as 'file_name'</pre> <p>다시 한번 시간기반 복구를 수행 합니다.</p> <ul style="list-style-type: none"> <li>● 리두 로그 파일에 있을 경우</li> </ul> <pre>alter database until time '2010-10-10:23:10:10' using backup controlfile</pre>

			<p>리두 로그 파일 직접 입력(redo01.log)</p> <ul style="list-style-type: none"> <li>● Archive log file에 있을 경우</li> </ul> <pre>alter database until time '2010-10-10:23:10:10' usnig backup controlfile</pre> <p>auto(아카이브 로그 파일 전체를 적용하는 의미)</p> <p>위와 같이 두 가지로 구분한 이유는 정확하게 어느 시점에 신규 테이블스페이스를 생성 하였는지도 중요하지만 기본적으로는 로그 스위치에 대한 정보도 반드시 봐야 합니다.</p> <p>역시 마지막으로 resetlogs 옵션을 이용해서 정상적으로 DB를 Open하시면 됩니다.</p> <pre>alter database open resetlogs</pre>
--	--	--	--

**5.5. 그 외 복구 방법은 “실전 오라클 백업과 복구 - 서진수 지음” 책을 이용하여 확인 하시기 바랍니다. 경우의 수가 많아서 문서로 기록하기에는 범위가 너무 많습니다.**

## 5.6. Export/Import 란

- expert 유틸리티는 데이터와 데이터베이스의 정의를 복사하고 이 내용을 오라클 내부 포맷의 바이너리 파일로 변환해서 OS 파일로 저장해 줍니다. import 유틸리티는 저장된 파일을 읽어서 다시 데이터베이스에 저장해주는 것입니다.
- Conventional Path Export와 Direct Path Export의 차이점
  - Conventional path export는 expert 명령어가 수행되면 expert client가 메모리에 evaluation buffer라는 곳을 만들고 DB Buffer cache에 있는 데이터를 evaluation buffer로 가져온 후 이곳이 다 차면 다시 디스크에 파일로 저장합니다.
  - Direct path expert는 Conventional path expert와 다르게 원본 테이블을 DB Buffer cache로 가져와서 파일로 바로 만들게 됩니다.
  - 여기에서 가장 큰 차이점은 Conventional Path는 여러 Process가 동시에 사용 할 수 있다는 장점이 있고 Direct Path는 다른 Process가 접근을 할 수 없습니다. 반대로 Direct Path는 Conventional path 보다 복잡한 구조가 아니기 때문에 속도가 빠릅니다.

## 5.7. Clone DB 만드는 과정을 설명하세요?

- 작업 순서
  1. 장애 상황 발생
    - A. 테스트용 테이블 스페이스 생성(여기서는 test01이라고 가정)
    - B. 테스트용 테이블 생성(여기에서는 gogak이라고 가정)
    - C. 테이블 삭제(시간을 확보됨)
  2. Clone DB용 파라미터 파일 생성(Pfile을 사용하겠습니다.)
    - A. db\_domain=""
    - B. db\_name=[새로 생성 할 인스턴스 이름 test02]
  3. 백업 데이터 파일을 복원(restore)
  4. 컨트롤 파일 재 생성
    - A. test01에서 alter database backup controlfile to trace as '경로 및 파일이름'
    - B. resetlogs 생성 스크립트를 만들어 주는데 기존과 다른 점은 REUSE를 SET로 변경 하고 TEST01을 TEST02로 변경 합니다.
    - C. 그 외 리두 로그 파일과 데이터 파일의 경로가 변경 해줍니다.(백업 데이터 파일 복원된 경로로 지정해주면 됩니다.)
  5. 장애 난 시점 직전까지 시간기반 복구로 테이블 복구(복구 방법 동일)
  6. Export 수행 후 운영 DB로 Import 수행 또는 DB Link를 이용한 CTAS
  7. 복구 확인
- 관련 내용은 "오라클 실전 백업과 복구 456p ~ 507p" 참조 바랍니다.

## 5.8. Flashback란?

- 사용자의 논리적인 오류를 아주 빠르게 복구해 낼 수 있는 방법 입니다. 하지만 물리적인 장애는 사용 할 수 없지만 사용자가 실수로 테이블을 삭제 하거나 DML에 의한 오류 등등을 복구 할 수 있다는 점은 반드시 기억 하셔야 합니다.
- Flashback 명령어의 3가지 Level
  - Row Level Flashback : 특정 row만 과거로 복구(commit 된 데이터만 복구가 가능)
  - Table Level Flashback : DROP(휴지통 기능), DML(UNDO)
  - Database Level Flashback : 시간기반 복구 DB전체를 복구, DB 전체를 스냅샷 하기 때문에 엄청난 용량을 사용하게 됨

## 5.9. Recovery Manger

- 오라클 데이터베이스에서 백업, 복원 및 복구 작업을 수행하는 유틸리티로 오라클 8i 이상에서 지원 합니다.

- 주요 기능으로는 아래와 같습니다.
  - 자주 실행하는 작업을 스크립트로 저장
  - 증분 Block 레벨 백업 기능 지원
  - 백업작업의 병렬화로 시간 절약
  - 사용되지 않는 Block 건너 뛰고 백업 수행
  - 백업 수행 중 훼손된 Block 감지
- 역시 이 부분은 "오라클 실전 백업과 복구 576P 부터 읽어 보시기 바랍니다."

## 6. 기타

FAST\_START\_MTTR\_TARGET

빠른 시작    Mean Time To Recovery

중요한 내용은 instance recovery를 하는 시간을 지정 할 수 있는데 만약 10초를 지정하면 10초안에 복구를 할 수 있는 데이터를 데이터 버퍼 캐쉬에 담아 놓는 다는 의미가 됩니다. 시간이 너무 적어도 문제지만 시간을 많이 설정하면 더 많은 공간을 확보해야 하기 때문에 좋다고 볼 수 없습니다.

SMON은 Instance recover만 하게 되고 Redo log file에서만 복구를 합니다.

## 7. 도와주신 분들

사이트
[교육] 서진수 쌤 오라클 취업반
[세미나] 김영조 쌤
[문서] 서진수 쌤 SQL 수업자료
[문서] 전문가로 가는 지름길 오라클 실습
[문서] 10g로 시작하는 오라클 SQL&PL/SQL
[문서] 초보자를 위한 오라클 10g
[문서] 오라클 실전 백업과 복구
[문서] 오라클 성능 고도화 원리화 해법 1
[문서] 오라클 성능 고도화 원리화 해법 2
[문서] Advanced Oracle Wait Interface in 10g
[구글머신] 무한 검색시대