**1. 관계형 데이터베이스의 개념**

관계형 데이터 모델은 1970년 6월 E.F.Codd 박사의 논문 “대량 공용 데이터베이스의 관계형 모델”(A Relational Model of Data for Large Shared Data Bank)에서 처음으로 데이터베이스 시스템용으로 관계형 모델 도입을 제안했다. 그 당시 더 많이 알려진 데이터 모델은 Hierarchical Data Model, Network Data Model, 심지어 단순 형태의 파일 데이터 구조였다.

RDBMS(Relational Database Management System)가 특히 사용의 용이, 구조상의 융통성 때문에 쉽게 대중화 되었고 게다가 총체적 해결을 제공하는 강력한 응용 프로그램 개발과 사용자 제품 군들을 RDBMS로 가능하게 해주는 업체들이 나타났고 그 중 Oracle이 대표적인 업체로 발전했다.

**1.1 관계형 모델의 구성요소**

1. 데이터를 저장하는 객체(object) 또는 관계(relation)들의 집합
2. 다른 관계를 생성하기 위해 관계에 가해지는 일련의 연산자 집합
3. 정확성과 일관성을 위한 데이터의 무결성(Integrity)

**1.2 관계형 데이터베이스 기능**

관계(relation)형 데이터베이스는 정보 저장을 위해 관계나 2자원 테이블을 이용한다.

1. 데이터의 저장을 관리한다.
2. 데이터에 대한 ACCESS을 통제한다.
3. 데이터를 검색 및 수정하기 위한 수단을 제공한다.

Database

Table Name : EMP Table Name : DEPT

EMPNO ENAME JOB DEPTNO DEPTNO DNAME LOC

------ ---------- --------- --------- --------- -------------- ---------

7839 KING PRESIDENT 10 10 ACCOUNTING NEW YORK

7698 BLAKE MANAGER 30 20 RESEARCH DALLAS

7782 CLARK MANAGER 10 30 SALES CHICAGO

7566 JONES MANAGER 20 40 OPERATIONS BOSTON

**1.3 관계형 데이터베이스의 정리**

1. E.F.Codd 박사는 1970년 데이터베이스 시스템용 관계형 모델을 제안
2. 제시한 관계형 모델은 관계형 데이터베이스 관리 시스템의 기본이 된다.
3. 관계형 모델링은 다음 구성요소를 포함하고 있다.
4. 객체(object) 또는 관계(relation)의 집합
5. 관계(relation)에 가해지는 연산의 집합
6. 정확성 및 일관성을 위한 데이터의 무결성
7. 관계형 데이터베이스는 2차원 테이블 형태로 구성된다.
8. 각 테이블은 Row와 Column으로 구성되어 있다.
9. 각 형의 데이터는 유일하다.
10. 각 column은 데이터 무결성을 유지한다.
11. SQL 명령어를 실행함으로 행들의 데이터를 조작 가능하다.

**1.4 관계 데이터베이스에서 사용하는 용어**

관계형 데이터베이스는 한 개 이상의 테이블을 가지고 있고 이 테이블은 RDBMS에서 기본 저장 구조입니다.

|  |
| --- |
| SQL> SELECT \* FROM emp; |

3 6 4

2 EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO

----- ---------- --------- ----- --------- --------- ----- ------

7839 KING PRESIDENT 17-NOV-81 5000 10

1 7698 BLAKE MANAGER 7839 01-MAY-81 2850 30

7782 CLARK MANAGER 7839 09-JUN-81 2450 10

7566 JONES MANAGER 7839 02-APR-81 2975 20

7654 MARTIN SALESMAN 7698 28-SEP-81 1250 1400 30

7499 ALLEN SALESMAN 7698 20-FEB-81 1600 300 30

7844 TURNER SALESMAN 7698 08-SEP-81 1500 0 30

7521 WARD SALESMAN 7698 22-FEB-81 1250 500 30

7902 FORD ANALYST 7566 03-DEC-81 3000 20

7369 SMITH CLERK 7902 17-DEC-80 800 20

7788 SCOTT ANALYST 7566 09-DEC-82 3000 20

7876 ADAMS CLERK 7788 12-JAN-83 1100 20

5

1. 특정 종업원에 대한 모든 데이터를 나타내는 단일 row 또는 tuple 입니다. 기본 키(primary key)에 의해 식별되어져야 합니다.
2. 기본 키(primary key)인 종업원 번호를 포함하는 열(column) 또는 속성(attribute)입니다. 종업원 번호는 EMP테이블에서 유일한(unique) 종업원을 식별합니다.
3. 키 값이 아닌 열 입니다. 열은 테이블에서 한 종류의 데이터를 나타냅니다.
4. 부서 번호를 포함하는 열은 외래 키(foreign key)입니다. 외래 키는 테이블 간에 서로 어떻게 관련되었는가를 정의합니다. 외래 키는 다른 테이블의 기본 키 또는 고유 키를 참조합니다.
5. 필드(field) 는 행과 열의 교차되는 곳에 있습니다.
6. 필드는 그 안에 값을 가지지 않을 수도 있습니다. 이것은 null value라 불립니다. EMP테이블에서 영업 사원인 종업원만이 COMM(commission)필드에서 값을 가집니다.

**1.5 다중 테이블 관계**

|  |
| --- |
| Table Name : EMP Table Name : DEPT  EMPNO ENAME JOB DEPTNO DEPTNO DNAME LOC  ----- ---------- --------- ------ ------ -------------- ---------  7839 KING PRESIDENT 10 10 ACCOUNTING NEW YORK  7698 BLAKE MANAGER 30 20 RESEARCH DALLAS  7782 CLARK MANAGER 10 30 SALES CHICAGO  7566 JONES MANAGER 20 40 OPERATIONS BOSTON  7654 MARTIN SALESMAN 30  7499 ALLEN SALESMAN 30 |

각 테이블은 하나의 ENTITY을 기술하는 데이터를 포함합니다. 데이터의 카테고리들은 각 테이블의 상단을 가로질러 LIST되며 개별 정보는 아래로 LIST됩니다. 테이블 포맷을 사용하여 정보를 사용하고, 이해하며, 쉽게 시각화 할 수 있습니다. 다른 ENTITY에 대한 데이터는 다른 테이블에 저장되기 때문에, 특정 문제에 대한 해답을 얻기 위해 둘 또는 보다 많은 테이블을 결합시킬 필요가 있을 수 있습니다. RDBMS는 외래 키를 사용하여 하나의 테이블에 다른 데이터를 관련시킬 수 있도록 해 줍니다. 외래 키는 동일 테이블 또는 다른 테이블에서의 기본 키를 참조하는 열 또는 열의 집합입니다. 한 테이블의 데이터를 다른 테이블의 데이터로 관련시키는 능력은 개별적으로 관리되는 정보의 단위들을 조직할 수 있도록 해 줍니다. 종업원 데이터는 부서 테이블과 분리된 테이블에 저장됨으로써 부서 데이터로부터 논리적으로 구분될 수 있습니다.

☞ Guidelines

* 기본 키와 외래 키

1. 기본 키에서 중복 값은 허용되지 않습니다.
2. 일반적으로 기본 키는 변경될 수 없습니다.
3. 외래 키는 데이터 값을 기초로 하며, 논리적이지 물리적이거나 포인터가 아닙니다.
4. 외래 키 값은 존재하는 기본 키 값 또는 고유 값과 일치해야 하거나, 또는 NULL이 될 수 있습니다.

**1.6 관계형 데이터베이스의 속성**

1. 관계형 데이터베이스에서는 테이블로의 액세스 루트를 지정하지 않으며 또 물리적으로 데이터가 어떻게 나열되는지 알 필요가 없다.
2. 데이터베이스를 이용하기 위해서는, 관계형 데이터베이스를 가동시킬 수 있는 ANSI(American National Standards Institute)표준 언어인 SQL(Structured Query Language)문을 실행해야 합니다. 이 언어는 관계(relation)를 조합, 분리 시킬 수 있는 일련의 큰 연산자 집합을 지니고 있습니다.
3. 데이터베이스는 SQL 문 사용으로 변경이 용이하다.
4. 데이터베이스는 데이터의 독립성을 보장한다.
5. 관계형 데이터베이스의 OBJECT의 종류는 다음과 같다

|  |  |
| --- | --- |
| OBJECT | 설 명 |
| TABLE | ROW와 COLUMN으로 구성된 기본적인 저장 단위 |
| VIEW | 하나 이상의 TABLE로부터 논리적으로 데이터를 분류한 부분집합 |
| INDEX | 포인터를 사용하여 행의 검색 속도를 향상 |
| SEQUENCE | 자동적으로 ORACLE SERVER이 유일 번호를 생성 |
| SYNONYM | 객체에 대체 이름을 부여 |
| PROGRAM UNIT | SQL또는 PL/SQL 문으로 작성한 PROCEDURE, FUNCTION, PACKAGE |

**1.7 SQL문장**

SQL Statement is entered

**SQL> select deptno,dname,loc from dept;**

Statement is sent to database

Database

Data is displayed

DEPTNO DNAME LOC

------ -------------- ---------

10 ACCOUNTING NEW YORK

20 RESEARCH DALLAS

30 SALES CHICAGO

40 OPERATIONS BOSTON

SQL로 서버와 통신이 가능하며 다음의 장점 등을 지니고 있습니다.

1. 효율적입니다.
2. 배우기 쉽고 사용하기 쉽습니다.
3. 기능적으로 완전합니다. SQL은 테이블에 있는 데이터를 정의, 검색, 조작할 수 있게 해 줍니다.

**2. ORACLE 관계형 데이터베이스 관리 시스템**

ORACLE SERVER의 특징은 관계형 구조에 근거한 모든 장점은 물론이고 프로그램 단위를 저장하고 실행 하도록 하는 PL/SQL 엔진을 통해 정보를 저장하고 관리할 수 있게 해줍니다. 서버는 최적화를 바탕으로 데이터를 읽어 들이는 사용자 옵션을 제공, 데이터베이스가 어떻게 ACCESS되고 사용되는지를 제어하는 보안 기능을 포함, Locking기법을 통해 데이터의 일관성과 보호 기능을 가능하도록 하는 것입니다.

ORACLE 어플리케이션은 ORACLE SERVER과 동일한 컴퓨터에서 실행될 수 있으며 다른 방식으로는 Oracle7 Server는 다른 하나의 시스템에서 수행되고 어플리케이션은 사용자의 로컬 시스템에서 실행되는 클라이언트-서버 형태로 운영할 수 있습니다. 이런 클라이언트-서버 환경에서는 광범위한 컴퓨팅 리소스가 사용될 수 있습니다.

최근에는 Oracle8이 발표되었는데 이는 ORACLE에 의해 개발된 첫 번째 객체 가능 데이터베이스 입니다. 이것은 새로운 ORDBMS를 지원하기 위하여 Oracle7의 데이터 모델링 능력을 확장합니다. Oracle8은 객체 지향 프로그래밍, 복잡한 데이터 타입, 복잡한 비즈니스 객체, 실 세계와 완전히 양립 가능한 새 엔진을 제공합니다. Oracle8은 많은 면에서 Oracle7을 확장합니다. 런타임 데이터 구조를 더 많이 공유하고, 더 큰 버퍼 캐쉬를 잡고, 제약 조건을 지연 가능케 하는 등 OLTP(Online Transaction Processing)의 기능성과 성능을 향상시킵니다. 데이터 웨어하우스 어플리케이션은 삽입, 갱신, 삭제의 병렬 수행, 분할, 병렬 인식 질의 최적화 같은 성능 향상 측면에서 장점이 있습니다. NCA(Network Computing Architecture)프레임 워크에서 작동 중인 Oracle8은 분산되고(distributed) 다중으로 고정된(multitiered) 클라이언트-서버와 웹 기반 어플리케이션을 지원합니다. Oracle8은 수십만의 동시 사용자를 조정할 수 있고521petabytes까지 지원하며 전통적으로 구조화된 데이터 뿐만 아니라 텍스트, 공간 이미지, 사운드, 비디오, 일련의 시간을 포함하는 어떤 타입의 데이터도 처리할 수 있습니다.

**2.1 완전한 ORACLE 솔루션**

ORACLE RDBMS는 ORACLE의 핵심 제품입니다. ORACLE SERVER과 데이터의 실제적인 사용, 모니터링, 유지 관리 등에 있어서 사용자를 돕기 위한 몇가지 툴들을 포함하고 있습니다. ORACLE 데이터 사전은 서버의 가장 중요한 구성요소로서 데이터베이스를 읽기 전용으로 참조할 수 있게 하는 VIEW과 테이블로 구성 되어 집니다. ORACLE의 RDBMS는 다음과 같은 일을 처리합니다.

1. 데이터 정의와 저장을 관리
2. 데이터 이용과 동시성 제어 및 제한
3. 백업과 복구 제공
4. SQL과 PL/SQL문 번역
5. 시스템 보안 유지

Application Developer/2000

HR Discover

Financials Designer/2000

Manufacturing

. . . . . .

SQL PL/SQL SQL\*Plus

Oracle7/8

Database Data Dictionary

Data Tables

**3. SQL, SQL\*Plus, PL/SQL**

SQL은 RDBMS를 사용하기 위해 ANSI에서 책정한 표준 언어로 Oracle 서버와 통신하기 위한 명령 언어이고 SQL\*Plus는 SQL 및 PL/SQL 문장을 인식하고 실행시켜주는 Oracle TOOL이다. PL/SQL Application Logic 추가하여 SQL을 확장한 Oracle의 절차적인 언어이다.

**3.1 SQL**

ORACLE SQL은 ANSI(American Standards Institute)와 ISO(Intermation Standards Organization) 표준을 따릅니다. ORACLE 회사는 SQL표준 위원회에 주요 인사를 적극적으로 포함시켜 ,변화, 발전하는 표준에 대한 미래 수용을 보증합니다. ANSI와ISO는 관계형 데이터베이스 표준 언어로 SQL을 채택했습니다.

**3.1.1 SQL 문장의 종류**

|  |  |  |
| --- | --- | --- |
| 구 분 | 문 장 | 설 명 |
| Retrive(Query) | SELECT | 데이터베이스로부터 데이터 검색 |
| DML(Data Manipulation Language) | INSERT  UPDATE  DELETE | 개별적으로 데이터베이스 테이블에서 새로운 행을 입력하고 기존의 행을 변경하고, 원치 않는 행을 제거 합니다. |
| DCL(Data Control Language) | CREATE  ALTER  DROP  RENAME  TRUNCATE | 데이터 구조를 생성, 변경, 제거합니다. |
| Transaction Control Language | COMMIT  ROLLBACK SAVEPOINT | DML명령문으로 만든 변경을 관리, 데이터 변경은 논리적 Transaction으로 함께 그룹화 될 수 있습니다. |
| DCL(Data Control Language) | GRANT  REVOKE | ORACLE 데이터베이스와 그 구조에서 액세스 권한을 제공하거나 제거합니다. |

**3.1.2 SQL 문장의 특징**

1. 사용하기가 쉬워 프로그래밍에 별로 경험이 없는 사용자는 물론 다양한 이용자들이 사용할 수 있다
2. 비절차적인 언어이다
3. 시스템 개발 및 유지 보수에 의해 소요되는 시간을 줄일 수 있다
4. 영어와 유사한 언어이다

**3.2 SQL\*Plus**

SQL\*Plus는 SQL, PL/SQL문장을 인식하고 실행을 위해 서버에 넘겨주는 Oracle TOOL로서 자체적인 명령 언어들을 갖고 있다

**3.2.1 SQL\*Plus 문장의 종류**

|  |  |
| --- | --- |
| 구 분 | 종 류 |
| Execution Commands | /, RUN, EXECUTE |
| Edit Commands | LIST, APPEND, CHANGE, DEL, INPUT, EDIT |
| Environment Commands | SET, SHOW, PAUSE |
| Report Format Commands | COLUMN, CLEAR, BREAK, COMPUTE, TTITLE, BTITLE |
| File Manipulation Commands | SAVE, GET, START, @, @@, SPOOL |
| Interactive Commands | DEFINE, UNDEFINE,PROMPT, ACCEPT, VARIABLE, PRINT |
| Database Access Commands | CONNECT, COPY, DISCONNECT |
| Miscellaneous Commands | SQLPLUS, EXIT, HELP, DESCRIBE, HOST, REMARK, RUNFORM, TIMING, WHENEVER |

**3.2.2 SQL\*Plus의 특징**

1. 수행하도록 입력한 문장들을 받아들인다
2. 파일에서 SQL 및 PL/SQL 입력 문을 받아들인다
3. 라인 편집기로 SQL 문장을 편집한다
4. 환경설정을 제어한다
5. 검색 결과를 보고서 형태로 형식 출력할 수 있도록 설정해 준다
6. 최종 사용자와 상호 작용한다
7. 원격지 데이터베이스(Remote database)를 액세스한다

**3.3 PL/SQL(Procedural Language/SQL)**

PL/SQL은 ORACLE 회사의 SQL에 대한 절차적 확장입니다. 객체 관계형 데이터베이스에 대한 표준 데이터 Access 언어입니다. PL/SQL은 데이터 캡슐화, 정보 숨김, 객체 지향 같은 현대적 소프트웨어 엔지니어링 기능을 제공하여 ORACLE SERVER과 툴 셋에 대한 최신식의 프로그래밍을 가능하게 합니다.

PL/SQL은 1970년대와 1980년대 사이에 설계된 프로그래밍 언어의 여러 향상된 기능을 포함합니다. PL/SQL을 강력한 트랜잭션 프로세싱 언어로 만듦으로써, SQL데이터 조작 및 질의 문장이 블록 구조와 절차적 코드 단위에 포함되도록 합니다. PL/SQL로 데이터를 처리하기 위해 PL/SQL제어 명령문을 사용할 수도 있고, ORACLE 데이터를 처리하기 위해 SQL명령문을 사용할 수도 있습니다.

**3.3.1 PL/SQL엔진과 Oracle Server**

PL/SQL은 별개의 Oracle 제품이 아니라 Oracle 서버와 다른 Oracle 툴에 이용되고 있는 프로그래밍 언어이다. PL/SQL의 블록은 Oracle 서버나 툴에 내장되는 PL/SQL 엔진에 전달되어 처리된다. 사용하는 엔진은 PL/SQL 이 수행되는 곳에 따라 다르다.

**가) Oracle 서버에서의 PL/SQL 엔진**

Pro\*프로그램, USER-EXIT, SQL\*Plus, 또는 Server Manager에서 PL/SQL 블록을 사용하면 Oracle 서버의 PL/SQL 엔진이 처리한다. 그리고 블록에 있는 SQL 을 별도의 문장으로 분리하여SQL 문 실행기로 보낸다. 이는 응용 프로그램의 블록을 한 번에 Oracle 서버에게 보낸다는 뜻이며 따라서 client/server 환경하에서 많은 성능 향상을 기대할 수 있다. 또한 데이터 베이스에 접속한 여러 응용 프로그램에서 stored subprogram 들을 참조할 수 있다.

**나) Oracle 툴에서의 PL/SQL**

Developer/2000을 포함한 많은 Oracle툴은 Oracle 서버에 있는 엔진과는 별도로 자체 PL/SQL 엔진을 갖고 있다. 이 엔진이 SQL 분장을 찾아서 Oracle 서버의 SQL문 실행기로 보내고, PL/SQL engine 은 응용 프로그램에 대해 지역적인 데이터를 처리한다. 이로써 Oracle 서버의 대한 작업량과 요구되는 메모리 커서의 수를 줄인다.

♣ 참고

Developer/2000 응용 프로그램의 부분으로 선언된 프로시저와 함수의 일반적인 구조는 동일하더라도 데이터베이스에 저장된 것과는 다르다. Stored Subprogram은 데이터베이스 객체이고 데이터 사전에 저장되며 여러 응용 프로그램이 사용할 수 있다. Application subprogram 그 응용 프로그램의 지역적인 PL/SQL 엔진에 블록을 전달한다. 작업은 서버 쪽이 아닌 응용 프로그램 쪽에서 수행된다.

**3.3.2 PL/SQL 문장의 종류**

기본 PL/SQL은 선언부, 실행부, 예외 처리부의 블록과 섹션으로 구성되어있다. 이들 중 실행부에서 사용할 수 있는 명령어들은 다음과 같다.

|  |  |
| --- | --- |
| 구 분 | 설 명 |
| IF | 조건에 따라 선택적으로 작업을 수행 |
| Basic LOOP | LOOP와 END LOOP사이의 문장을 반복 수행 |
| FOR loop | 반복되는 수를 정하여 문장을 반복 수행 |
| WHILE loop | 제어 조건이 TRUE가 아닐 때까지 문장을 반복 수행 |

**3.3.3 PL/SQL의 특징**

1) SQL로는 얻을 수 없는 PL/SQL의 절차적 프로그래밍 기능

2) 모듈화된 프로그램 개발

1. 블록 내에서 논리적으로 관련된 문장들의 그룹화
2. 강력한 프로그램을 작성하기 위해 서브 블록들을 큰 블록에 포함
3. 복잡한 문제에 대한 프로그래밍이 적절히 나뉘어진 모듈들의 집합으로 구성

3) CURSOR, EXCEPTION

1. 변수, 상수 등을 선언하고 SQL과 절차적인 프로그램에서 사용
2. 데이터베이스의 테이블과 Record를 기반으로 하는 dynamic한 변수 선언이 가능

4) 절차적 언어 구조로 된 프로그램 작성

1. 조건에 따라 일련의 문장을 실행 (IF)
2. 루프에서 반복적으로 일련의 문장을 실행 (LOOP)
3. Explicit Cursor를 이용한 Multi-row 질의 처리

5) ERROR 처리

1. Exception 처리 루틴을 이용하여 ORACLE8 서버 에러를 처리함
2. 사용자 정의 에러를 선언하고 Exception 처리 루틴으로 처리 가능 함