DATABASE DESIGN & IMPLEMENTATION

OBJECTIVES

- CONSTRUCT AND EXECUTE JOINS BETWEEN TABLES
- CONSTRUCT AND EXECUTE THE FOLLOWING JOINS USING ANSI-99 SQL JOIN SYNTAX
- NATURAL JOIN
- CROSS JOIN
- EQUIJOIN
- NON EQUIJOIN APPLY THE RULE FOR USING TABLE ALIASES IN A JOIN STATEMENT

ANSI

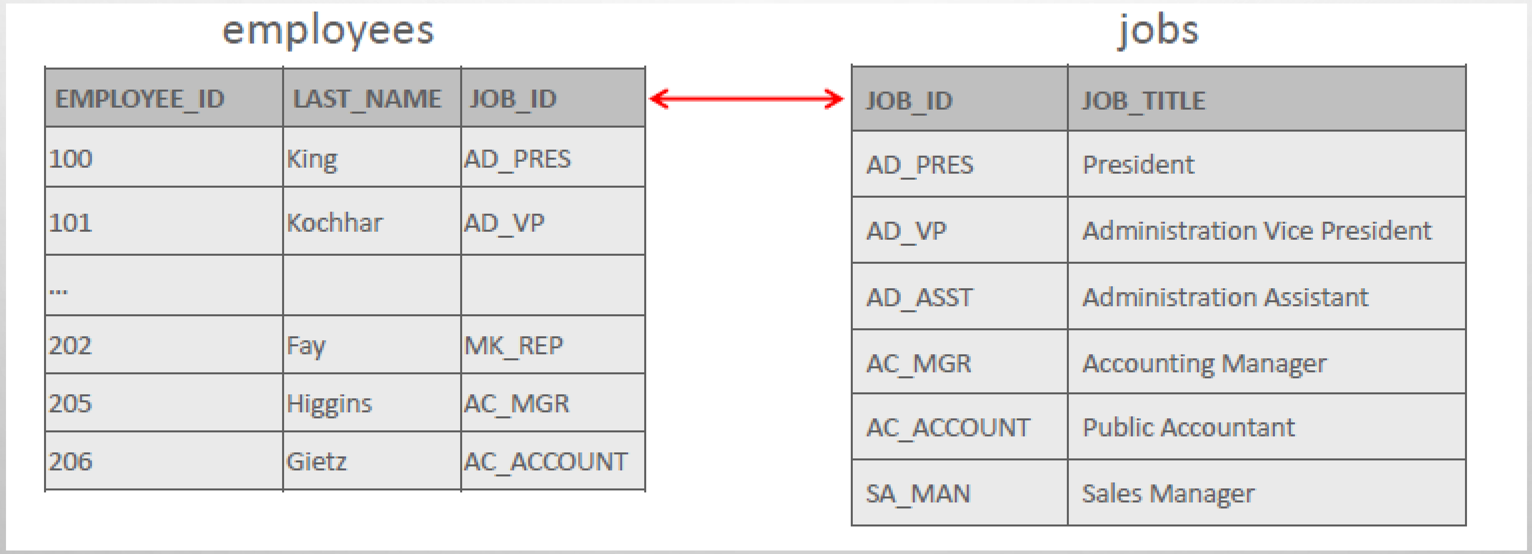
- AMERICAN NATIONAL STANDARDS INSTITUTE (ANSI)
- FOUNDED IN 1918, ANSI IS A PRIVATE, NON-PROFIT ORGANISATION THAT ADMINISTERS AND COORDINATES THE U.S VOLUNTARY STANDARDISATION AND CONFORMITY ASSESSMENT SYSTEM.
- STRUCTURED QUERY LANGUAGE (SQL) IS THE INFORMATION PROCESSING INDUSTRY STANDARD LANGUAGE OF RELATIONAL DATABASE MANAGEMENT SYSTEMS (RDBMS).
- THE LANGUAGE WAS ORIGINALLY DESIGNED BY IBM IN THE MID 1970'S, CAME INTO WIDESPREAD USE IN THE EARLY 1980'S, AND BECAME AN INDUSTRY STANDARD IN 1986 WHEN IT WAS ADOPTED BY ANSI.
- THREE STANDARDS EACH NAMED FOR THE YEAR THEY WERE FIRST PROPOSED, EACH ONE BUILDING ON THE PREVIOUS ONE.
- ANSI-86, ANSI-92 AND ANSI-99

NATURAL JOIN

- A SQL JOIN CLAUSE COMBINES FIELDS FROM 2 (OR MORE) TABLES IN A RELATIONAL DATABASE.
- A NATURAL JOIN IS BASED ON ALL COLUMNS IN TWO TABLES THAT HAVE THE SAME NAME AND SELECTS ROWS FROM THE TWO TABLES THAT HAVE EQUAL VALUES IN ALL MATCHED COLUMNS.

NATURAL JOIN

- THE EMPLOYEES TABLE HAS A JOB_ID COLUMN.
- THIS IS A REFERENCE TO THE COLUMN OF THE SAME NAME IN THE JOBS TABLE.



employees

EMPLOYEE_ID	LAST_NAME	JOB_ID
100	King	AD_PRES
101	Kochhar	AD_VP
...		
202	Fay	MK_REP
205	Higgins	AC_MGR
206	Gietz	AC_ACCOUNT

jobs

JOB_ID	JOB_TITLE
AD_PRES	President
AD_VP	Administration Vice President
AD_ASST	Administration Assistant
AC_MGR	Accounting Manager
AC_ACCOUNT	Public Accountant
SA_MAN	Sales Manager

NATURAL JOIN

- AS SHOWN IN THE SAMPLE CODE, WHEN USING A NATURAL JOIN, IT IS POSSIBLE TO JOIN THE TABLES WITHOUT HAVING TO EXPLICITLY SPECIFY THE COLUMN NAMES IN THE CORRESPONDING TABLE.
- HOWEVER THE NAMES AND DATA TYPES OF BOTH COLUMNS MUST BE THE SAME.

```
SELECT FIRST_NAME, LAST_NAME, JOB_ID, JOB_TITLE

FROM EMPLOYEES NATURAL JOIN JOBS

WHERE DEPARTMENT_ID > 80;
```

NATURAL JOIN

- THE DEPARTMENTS AND LOCATIONS TABLE BOTH HAVE A COLUMN, LOCATION_ID, WHICH IS USED TO JOIN THE TWO TABLES.

```
SELECT DEPARTMENT_NAME, CITY
FROM DEPARTMENTS NATURAL JOIN LOCATIONS;
```

- NOTICE THAT THE NATURAL JOIN COLUMN DOES NOT HAVE TO APPEAR IN THE SELECT CLAUSE.

USING CLAUSE

- IN A NATURAL JOIN, IF THE TABLES HAVE COLUMNS WITH THE SAME NAMES BUT DIFFERENT DATA TYPES, THE JOIN CAUSES AN ERROR.
- TO AVOID THIS SITUATION, THE JOIN CLAUSE CAN BE MODIFIED WITH A USING CLAUSE.
- THE USING CLAUSE SPECIFIES THE COLUMNS THAT SHOULD BE USED FOR THE JOIN.
- A USING CLAUSE IS OFTEN PREFERRED TO A NATURAL JOIN EVEN WHEN THE COLUMNS HAVE THE SAME DATA TYPE AS WELL AS THE SAME NAME, BECAUSE IT CLEARLY STATES EXACTLY WHICH JOIN COLUMN IS BEING USED.

USING CLAUSE

- THE COLUMNS REFERENCED IN THE USING CLAUSE SHOULD NOT HAVE A QUALIFIER (TABLE NAME OR ALIAS).

```
SELECT FIRST_NAME, LAST_NAME, DEPARTMENT_NAME

FROM EMPLOYEES

JOIN DEPARTMENTS

USING (DEPARTMENT_ID);
```

USING CLAUSE

- THE USING CLAUSE ALLOWS US TO USE WHERE TO RESTRICT ROWS FROM ONE OR BOTH TABLES;

```
SELECT FIRST_NAME, LAST_NAME, DEPARTMENT_NAME
FROM EMPLOYEES
JOIN DEPARTMENTS
USING (DEPARTMENT_ID)
WHERE LAST_NAME = ‘HIGGINS’;
```

ON CLAUSE

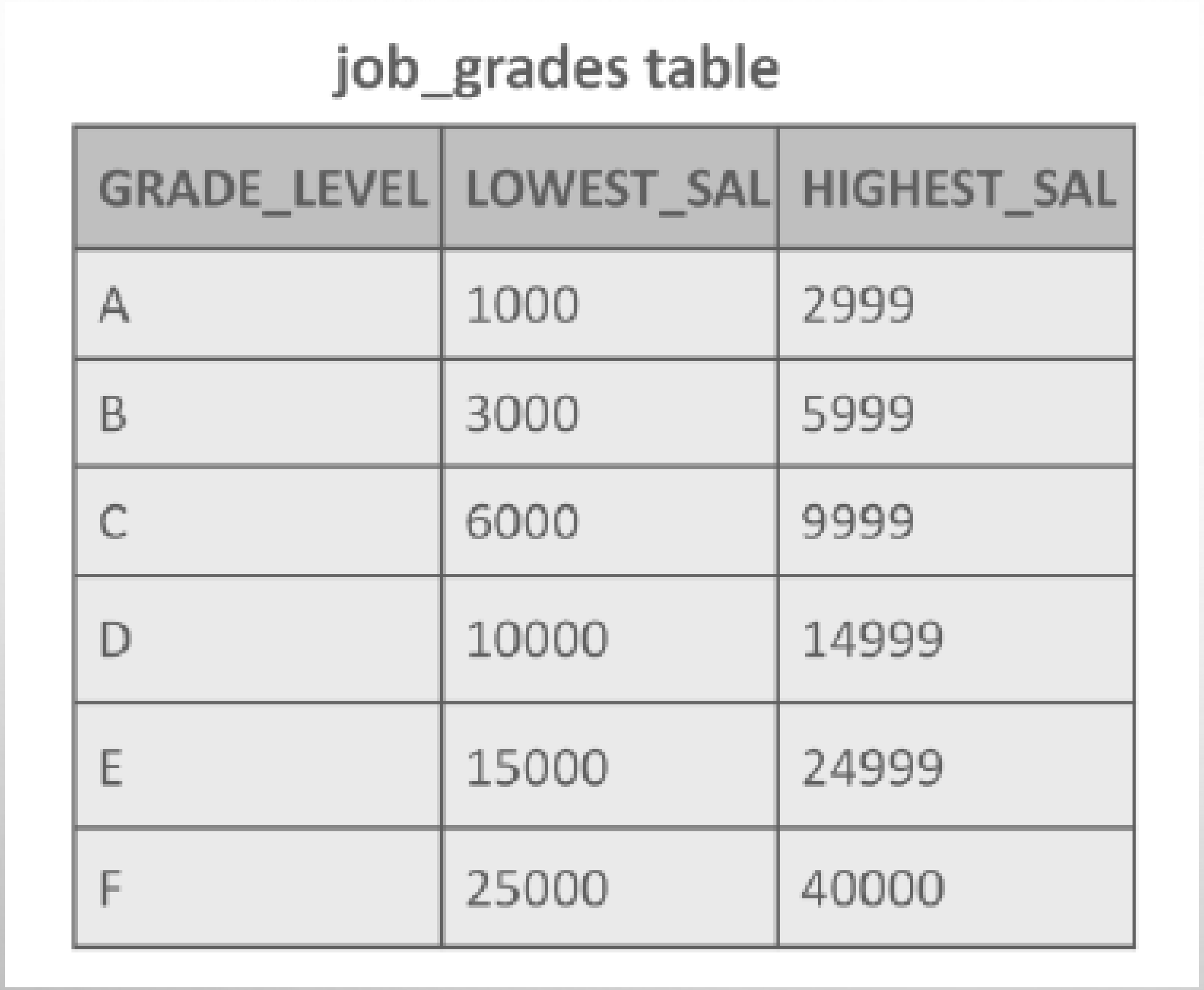
- WHAT IF THE COLUMNS TO BE JOINED HAVE DIFFERENT NAMES, OR IF THE JOIN USES NON-EQUAL COMPARISON OPERATORS SUCH AS <, >, OR BETWEEN?
- WE CAN'T USE USING SO INSTEAD WE USE ON
- THIS ALLOWS A GREATER VARIETY OF JOIN CONDITIONS TO BE SPECIFIED
- THE ON CLAUSE ALSO ALLOWS US TO USE WHERE TO RESTRICT ROWS FROM ONE OR BOTH TABLES

```
SELECT LAST_NAME, JOB_TITLE

FROM EMPLOYEES JOIN JOBS

ON (JOB_ID = JOB_NO);
```

ON CLAUSE

- A JOIN ON CLAUSE IS REQUIRED WHEN THE COMMON COLUMNS HAVE DIFFERENT NAMES IN THE TWO TABLES
- WHEN USING THE ON CLAUSE ON COLUMNS WITH THE SAME NAME IN BOTH TABLES, YOU NEED TO ADD A QUALIFIER

```
SELECT LAST_NAME, JOB_TITLE
FROM EMPLOYEES E JOIN JOBS J
ON (E.JOB_ID = J.JOB_ID)
WHERE LAST_NAME LIKE ‘H%’;
```



job_grades table

GRADE_LEVEL	LOWEST_SAL	HIGHEST_SAL
A	1000	2999
B	3000	5999
C	6000	9999
D	10000	14999
E	15000	24999
F	25000	40000

ON CLAUSE WITH NON-EQUALITY OPERATOR

- SOMETIMES YOU MAY NEED TO RETRIEVE DATA FROM A TABLE THAT HAS NO CORRESPONDING COLUMN IN ANOTHER TABLE.
- SUPPOSE WE WANT TO KNOW THE GRADE_LEVEL FOR EACH EMPLOYEES SALARY.
- THE JOB_GRADES TABLE DOES NOT HAVE A COMMON COLUMN WITH THE EMPLOYEES TABLE.
- USING AN ON CLAUSE ALLOWS US TO JOIN THE TWO TABLES.

ON CLAUSE WITH NON-EQUALITY OPERATOR

```
SELECT LAST_NAME, SALARY, GRADE_LEVEL, LOWEST_SAL,
HIGHEST_SAL

FROM EMPLOYEES

JOIN JOB_GRADES

ON (SALARY BETWEEN LOWEST_SAL AND HIGHEST_SAL);
```

JOINING THREE TABLES

- BOTH USING AND ON CAN BE USED TO JOIN THREE OR MORE TABLES.
- SUPPOSE WE NEED A REPORT OF OUR EMPLOYEES, THEIR DEPARTMENT, AND THE CITY WHERE THE DEPARTMENT IS LOCATED?
- WE NEED TO JOIN THREE TABLES: EMPLOYEES, DEPARTMENTS AND LOCATIONS.

```
SELECT LAST_NAME, DEPARTMENT_NAME AS “DEPARTMENT”, CITY
FROM EMPLOYEES JOIN JOB_GRADES USING (DEPARTMENT_ID)
JOIN LOCATIONS USING (LOCATION_ID);
```

PRACTICE

- JOIN THE LOCATIONS AND DEPARTMENTS TABLE USING THE LOCATION_ID COLUMN, LIMIT THE RESULTS TO LOCATION 1400 ONLY.
- JOIN D_PLAY_LIST_ITEMS, AND D_TRACK_LISTINGS USING SONG_ID AND THEN JOIN TO D_CDS USING CD_NUMBER. INCLUDE IN THE OUTPUT THE SONG_ID, CD_NUMBER, TITLE AND COMMENTS GIVING EACH AN APPROPRIATE NAME.
- DISPLAY THE CITY, DEPARTMENT NAME, LOCATION ID AND DEPARTMENT ID FOR DEPARTMENTS 10, 20, AND 30 FOR THE CITY OF SEATTLE.
- LOCATIONS TABLE HAS THE CITY STORED AND THE LOCATION_ID, DEPARTMENTS TABLE HAS THE DEPARTMENT_NAME, DEPARTMENT_ID AND LOCATION_ID