1. sqlDB를 사용하여 ifProc 이라는 이름의 procedure를 생성하고 var1에 저장된 값이 100인지 확인

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

+---------------+

| 100입니다. |

+---------------+

| 100입니다. |

+---------------+

1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

2. employees를 사용하여 ifProc2 라는 이름의 proceudre를 생성. hireDATE(DATE) curDATE(DATE) days(INT) 변수 선언, hire\_date에서 emp\_no가 10001인 경우에 대해서 hireDATE 대입. curDATE 오늘 날짜 설정, days는 DATEDIFF 함수를 이용하여 계산.(days/365) >= 5 일 경우 축하 메시지. 아닐 경우 다른 메시지

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

+-------------------------------------------------------------------------------+

| CONCAT ('입사한지', days, '일이나 지났습니다. 축하합니다.!') |

+-------------------------------------------------------------------------------+

| 입사한지11822일이나 지났습니다. 축하합니다.! |

+-------------------------------------------------------------------------------+

1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

3. 점수에 따라서 학점을 주는 코드 생성. IF 와 CASE 를 이용한 2가지 경우에 대해서 생성.

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

+----------------------------------+-----------------------------+

| CONCAT('취득점수==>', point) | CONCAT('학점==>', credit) |

+----------------------------------+-----------------------------+

| 취득점수==>77 | 학점==>C |

+----------------------------------+-----------------------------+

1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

+----------------------------------+-----------------------------+

| CONCAT('취득점수==>', point) | CONCAT('학점==>', credit) |

+----------------------------------+-----------------------------+

| 취득점수==>77 | 학점==>C |

+----------------------------------+-----------------------------+

1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

3. WHILE을 이용하여 1~100까지의 합 계산

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

+------+

| hap |

+------+

| 5050 |

+------+

1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

4. Iterate 를 이용하여 값이 100보다 작은 동안 7의 배수가 아닌 경우에 대해서 합계 계산

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

+------+

| hap |

+------+

| 1029 |

+------+

1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)