Updatable views

(From the MySQL manual)

Some views are updatable. That is, you can use them in statements such as `UPDATE`, `DELETE`, or `INSERT` to update the contents of the underlying table. There are also certain other constructs that make a view nonupdatable. To be more specific, a view is not updatable if it contains any of the following:

- Aggregate functions (`SUM()`, `MIN()`, `MAX()`, `COUNT()`, and so forth)
- `DISTINCT`
- `GROUP BY`
- `HAVING`
- `UNION` or `UNION ALL`
- Subquery in the select list
- Certain joins (see additional join discussion later in this section)
- Nonupdatable view in the `FROM` clause
- A subquery in the `WHERE` clause that refers to a table in the `FROM` clause
- Refers only to literal values (in this case, there is no underlying table to update)
- Uses `ALGORITHM = TEMPTABLE` (use of a temporary table always makes a view nonupdatable)
- Multiple references to any column of a base table.

Insertable views

- All of the criteria for Updatable views
- There must be no duplicate view column names.
- The view must contain all columns in the base table that do not have a default value.
- The view columns must be simple column references and not derived columns. A derived column is one that is not a simple column reference but is derived from an expression.

Multiple Table views

- It is sometimes possible for a multiple-table view to be updatable, assuming that it can be processed with the `MERGE` algorithm. For this to work, the view must use an inner join (not an outer join or a `UNION`). Also, only a single table in the view definition can be updated, so the `SET` clause must name only columns from one of the tables in the view. Views that use `UNION ALL` are disallowed even though they might be **theoretically updatable, because the implementation uses temporary tables to** process them.
- For a multiple-table updatable view, `INSERT` can work if it inserts into a single table.

Examples:

The following shows an insertable view and a non-insertable view due to the base table not having a default value for a field not in the view.

```
mysql> describe class;
+-------------+----------+------+-----+---------+-------+
| Field       | Type     | Null | Key | Default | Extra |
+-------------+----------+------+-----+---------+-------+
| ClassID     | char(20) | NO   | PRI |         |       |
| TimeOffered | char(10) | YES  |     | NULL    |       |
+-------------+----------+------+-----+---------+-------+
2 rows in set (0.22 sec)

mysql> create view c as
    -> select ClassID from Class;
Query OK, 0 rows affected (0.03 sec)

mysql> insert into c values ("CS575");
Query OK, 1 row affected (0.03 sec)

mysql>

mysql> describe Student;
+-------------+----------+------+-----+---------+-------+
| Field       | Type     | Null | Key | Default | Extra |
+-------------+----------+------+-----+---------+-------+
| StudentID   | int(11)  | NO   | PRI | 0       |       |
| StudentName | char(20) | NO   |     |         |       |
+-------------+----------+------+-----+---------+-------+
2 rows in set (0.00 sec)

mysql> CREATE VIEW v as
    -> select StudentID from Student;
Query OK, 0 rows affected (0.03 sec)

mysql> insert into v values (1234);
ERROR 1423 (HY000): Field of view 'cs430.v' underlying table doesn't have a defa
ult value
mysql>
```

The following is exercise 3.19 from the textbook.
Briefly answer the following questions based on this schema:

Emp(*eid:* integer, *ename:* string, *age:* integer, *salary:* real)
Works(*eid:* integer, *did:* integer, *pct time:* integer)
Dept(*did:* integer, *budget:* real, *managerid:* integer)

1. Suppose you have a view SeniorEmp defined as follows:

```
CREATE VIEW SeniorEmp (sname, sage, salary)
        AS SELECT E.ename, E.age, E.salary
        FROM Emp E
                WHERE E.age > 50
```

Explain what the system will do to process the following query:

```
SELECT S.sname
        FROM SeniorEmp S
        WHERE S.salary > 100,000
```

The system will do the following:

```
SELECT S.name
        FROM ( SELECT E.ename AS name, E.age, E.salary
                FROM Emp E
                WHERE E.age > 50 ) AS S
        WHERE S.salary > 100000
```

2. Give an example of a view on Emp that could be automatically updated by updating Emp.

The following view on Emp can be updated automatically by updating Emp:

```
CREATE VIEW SeniorEmp (eid, name, age, salary)
        AS SELECT E.eid, E.ename, E.age, E.salary
        FROM Emp E
        WHERE E.age > 50
```

3. Give an example of a view on Emp that would be impossible to update (automatically) and explain why your example presents the update problem that it does.

The following view cannot be updated automatically because it is not clear which employee records will be affected by a given update:

```
CREATE VIEW AvgSalaryByAge (age, avgSalary)
        AS SELECT E.eid, AVG (E.salary)
        FROM Emp E
        GROUP BY E.age
```