APPENDIX 10.A: SQL:2016 SYNTAX SUMMARY

This appendix summarizes the SQL:2016 syntax for the CREATE VIEW statement presented in Chapter 10 and a simple companion statement (DROP VIEW). The conventions used in the syntax notation are identical to those used at the end of Chapter 3.

۲

.....

CREATE VIEW Statement

```
CREATE VIEW ViewName [ ( ColumnName* ) ]
AS <Select-Statement>
[ WITH CHECK OPTION ]
<Select-Statement>: -- defined in Chapter 4 and extended in Chapter 9
```

DROP VIEW Statement

DROP VIEW ViewName [{ CASCADE | RESTRICT }]

- -- CASCADE deletes the view and any views that use its definition.
- -- RESTRICT means that the view is not deleted if any views use its definition.

۲

 (\bullet)

APPENDIX 10.B: RULES FOR UPDATABLE JOIN VIEWS IN ORACLE

۲

In recent Oracle versions (9i to 12c), a join view that contains one or more tables or views in its defining FROM clause can be updatable. Fundamental to updatable join views is the concept of a key preserving table. A join view preserves a table if every candidate key of the table can be a candidate key of the join result table. This statement means that the rows of an updatable join view can be mapped in a 1-1 manner with each key preserved table. In a join involving a 1-M relationship, the child table could be key preserved because each child row is associated with at most one view row.

Using the definition of a key preserving table, a join view is updatable if it satisfies the following conditions:

- It does not contain the DISTINCT keyword, the GROUP BY clause, aggregation functions, or set operations (UNION, MINUS, and INTERSECT).
- It contains at least one key preserving table.
- The CREATE VIEW statement does not contain the WITH CHECK OPTION.

An updatable join view supports insert, update, and delete operations on one underlying table per manipulation statement. The updatable table is the key preserving table. An UPDATE statement can modify (in the SET clause) only columns of one key preserving table. An INSERT statement can add a row in the key preserving table as long as the view includes all required columns without default values. An INSERT statement cannot contain columns from nonkey preserving tables. Rows can be deleted as long as the join view contains only one key preserving table. Join views with more than one key preserving table do not support DELETE statements.

To improve your understanding of updatable join views, let us consider an example along with operations supported by the view. Example 10.B1 involves an updatable join view between the *Course* and *Offering* tables. The *Offering* table is key preserving as each view row maps to exactly one *Offering* row as the view contains *Offering*. *OfferNo*, the primary key of the *Offering* table. The *Course-Offering-View* is updatable as it contains a key preserving table, omits features that eliminate updatability (DIS-TINCT, GROUP BY, aggregation functions, and set operators), and does not contain the WITH CHECK clause.

Example 10.B1 (Oracle)

Updatable Join View

Create an updatable join view between the *Course* and the *Offering* tables. This view supports insert operations on both the key preserving table (*Offering*).

```
CREATE VIEW Course_Offering_View AS
SELECT Course.CourseNo, CrsDesc, CrsUnits,
Offering.OfferNo, OffTerm, OffYear,
OffLocation, OffTime, FacNo, OffDays
OffLimit, OffNumEnrolled
FROM Course INNER JOIN Offering
ON Course.CourseNo = Offering.CourseNo
```

The Course-Offering-View supports a limited set of operations that map to operations on the key preserving table. The following list summarizes operations supported by the *Course-Offering-View*.

- Insert operations using the Offering columns result in new Offering rows.
- Insert operations using Course columns are rejected as illegal operations.

- Update operations involving *Offering* columns change the columns in the associated *Offering* rows.
- Delete operations remove corresponding *Offering* rows.

The operations supported by Oracle updatable join views are less than Microsoft Access 1-M updatable queries. 1-M updatable queries support operations on both parent and child tables, while Oracle updatable join views support operations only on the key preserving table, typically the child table. Oracle provides INSTEAD OF triggers to support more operations on multiple table views as presented in Chapter 11.

۲

۲

()

APPENDIX 10.C: SOLUTIONS FOR QUERY FORMULATION ERRORS

۲

Appendix 10.C explains formulation errors in the queries for the Registration Form in Examples 10.36 and 10.37. Examples 10.C1 and 10.C2 show the original queries in Examples 10.36 and 10.37, corrected queries, and explanations about formulation errors.

Example 10.C1 (Access)

Query for the Main Form of the Registration Form

The SELECT statement contains two formulation errors (underlined) covered in Table 10-6. Since this query should support insert operations on the *Registration* table, join columns in the result should be foreign keys in the *Registration* table. Thus, *Registration.StdNo* should replace *Student.StdNo* in the result list. Using *Student.StdNo* instead of *Registration.StdNo*, the query does not support insert operations on the *Registration* table. The second error involves the GROUP BY clause. You should remove the GROUP BY clause as GROUP BY makes a query read-only.

```
-- Query with underlined formulation errors
SELECT RegNo, RegTerm, RegYear, RegDate,
        <u>Student.StdNo</u>, RegStatus, StdFirstName,
        StdLastName, StdClass, StdCity, StdState
FROM Registration INNER JOIN Student
        ON Registration.StdNo = Student.StdNo
        <u>GROUP BY RegNo, RegTerm, RegYear, RegDate,</u>
        <u>Student.StdNo, RegStatus, StdFirstName,
        StdLastName, StdClass, StdCity, StdState</u>
-- Corrected query with underlined corrections
SELECT RegNo, RegTerm, RegYear, RegDate,
        <u>Registration.StdNo</u>, RegStatus, StdFirstName,
        StdLastName, StdClass, StdCity, StdState
FROM Registration INNER JOIN Student
        ON Registration.StdNo = Student.StdNo
```

Example 10.C2 (Access)

()

Query for the Subform of the Registration Form

The SELECT statement contains three formulation errors (underlined) covered in Table 10-6. For the first error, the query omits the linking column (*Enrollment.RegNo*). Without the linking column, the subform displays all *Enrollment* rows, not the rows associated with the *Registration* row displayed in the main form. The linking column can appear anywhere in the subform query although placing it first makes it prominent. Note that the Microsoft Access requires the linking column in the query result even if the linking column does not appear in the subform. For the second error, the query should use the *Enrollment.OfferNo* (foreign key), not *Offering.OfferNo* (primary key). With *Offering.OfferNo* instead of *Enrollment.OfferNo*, the query is read-only for the *Enrollment* table. For the third error, the <u>Registration</u> table should not appear in the subform as it is the parent table in the main form.

-- Query with underlined formulation errors SELECT ____, Offering.OfferNo, Offering.CourseNo, OffTime, OffLocation, OffTerm, OffYear, Offering.FacNo, FacFirstName, FacLastName, CrsDesc, CrsUnits

Chapter 10 Application Development with Views 439

```
FROM ( ( Enrollment INNER JOIN Offering
           ON Enrollment.OfferNo = Offering.OfferNo )
      INNER JOIN Faculty
           ON Faculty.FacNo = Offering.FacNo )
       INNER JOIN Course
           ON Course.CourseNo = Offering.CourseNo )
       INNER JOIN Registration
           ON Registration.RegNo = Enrollment.RegNo
-- Corrected query with underlined corrections
SELECT <u>RegNo</u>, <u>Enrollment.OfferNo</u>, Offering.CourseNo,
        OffTime, OffLocation, OffTerm, OffYear,
        Offering.FacNo, FacFirstName, FacLastName,
        CrsDesc, CrsUnits
 \ensuremath{\mathsf{FROM}} ( ( \ensuremath{\mathsf{Enrollment}} \ensuremath{\mathsf{INNER}} JOIN Offering
           ON Enrollment.OfferNo = Offering.OfferNo )
       INNER JOIN Faculty
           ON Faculty.FacNo = Offering.FacNo )
       INNER JOIN Course
           ON Course.CourseNo = Offering.CourseNo
```

۲

۲

۲