```
mysql>
mysql> SELECT * FROM copy_d_songs;
+----+-------------+----------+----------+-----------+
| id | title       | duration | artist   | type_code |
+----+-------------+----------+----------+-----------+
|  1 | Song One    |        3 | Artist A | New Age   |
|  2 | Song Two    |        4 | Artist B | Rock      |
|  3 | Song Three  |        5 | Artist C | New Age   |
|  4 | Song Four   |        2 | Artist D | Jazz      |
|  1 | Song One    |        3 | Artist A | New Age   |
|  2 | Song Two    |        4 | Artist B | Rock      |
|  3 | Song Three  |        5 | Artist C | New Age   |
|  4 | Song Four   |        2 | Artist D | Jazz      |
| 88 | Mello Jello |        2 | The What | 4         |
+----+-------------+----------+----------+-----------+
9 rows in set (0.00 sec)

mysql>
mysql> -- 5. Create a view read_copy_d_cds with a WHERE clause and READ ONLY option
mysql> CREATE OR REPLACE VIEW read_copy_d_cds AS
    -> SELECT * FROM copy_d_cds
    -> WHERE year = 2000;
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> -- 6. DELETE operation on read_copy_d_cds view
mysql> DELETE FROM copy_d_cds WHERE cd_number = 90 AND year = 2000;
Query OK, 1 row affected (0.01 sec)

mysql>
mysql> -- 7. Modify read_copy_d_cds with CHECK OPTION
mysql> CREATE OR REPLACE VIEW read_copy_d_cds AS
    -> SELECT * FROM copy_d_cds
    -> WHERE year = 2000
    -> WITH CHECK OPTION;
Query OK, 0 rows affected (0.01 sec)
```

```
mysql>
mysql> SELECT * FROM read_copy_d_cds;
+-----------+--------+------+
| cd_number | title  | year |
+-----------+--------+------+
|         1 | CD One | 2000 |
+-----------+--------+------+
1 row in set (0.00 sec)

mysql>
mysql> -- 8. DELETE CDs of year 2000 from read_copy_d_cds
mysql> DELETE FROM copy_d_cds WHERE year = 2000;
Query OK, 1 row affected (0.01 sec)

mysql>
mysql> -- 9. DELETE cd_number 90 from read_copy_d_cds
mysql> DELETE FROM copy_d_cds WHERE cd_number = 90 AND year = 2000;
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> -- 10. DELETE year 2001 records from read_copy_d_cds
mysql> DELETE FROM copy_d_cds WHERE year = 2001;
Query OK, 1 row affected (0.01 sec)

mysql>
mysql> -- 11. SELECT from copy_d_cds base table to check deletions
mysql> SELECT * FROM copy_d_cds;
Empty set (0.00 sec)

mysql>
mysql> -- Managing Views
mysql>
mysql> -- 1. Create a view from the copy_d_songs table called view_copy
mysql> CREATE OR REPLACE VIEW view_copy_d_songs AS
    -> SELECT title, artist
    -> FROM copy_d_songs;
Query OK, 0 rows affected (0.01 sec)

mysql>
```

```
mysql>
mysql> SELECT * FROM view_copy_d_songs;
+-------------+----------+
| title       | artist   |
+-------------+----------+
| Song One    | Artist A |
| Song Two    | Artist B |
| Song Three  | Artist C |
| Song Four   | Artist D |
| Song One    | Artist A |
| Song Two    | Artist B |
| Song Three  | Artist C |
| Song Four   | Artist D |
| Mello Jello | The What |
+-------------+----------+
9 rows in set (0.00 sec)

mysql>
mysql> -- 2. Drop view_copy_d_songs and verify deletion
mysql> DROP VIEW IF EXISTS view_copy_d_songs;
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> SELECT * FROM information_schema.views WHERE table_name = 'view_copy_d_songs';
Empty set (0.01 sec)
```

```
mysql>
mysql> -- 4. Construct an inline view that lists the last name, salary, department ID, and maximum salary by department
mysql> SELECT e.last_name, e.salary, e.department_id, d.max_salary
    -> FROM employees e
    -> JOIN (SELECT department_id, MAX(salary) AS max_salary
    ->       FROM employees
    ->       GROUP BY department_id) d
    -> ON e.department_id = d.department_id;
ERROR 1054 (42S22): Unknown column 'e.last_name' in 'field list'
mysql>
mysql> -- 5. Return the staff members of Global Fast Foods ranked by salary from lowest to highest
mysql> SELECT last_name, salary
    -> FROM employees
    -> WHERE company = 'Global Fast Foods'
    -> ORDER BY salary ASC;
ERROR 1054 (42S22): Unknown column 'last_name' in 'field list'
mysql>
mysql> -- Indexes and Synonyms
mysql>
mysql> -- 4. Create a nonunique index for cd_number in d_track_listings
mysql> CREATE INDEX idx_cd_number ON d_track_listings (cd_number);
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
mysql> -- 5. Display indexes and uniqueness in d_songs
mysql> SELECT index_name, non_unique
    -> FROM information_schema.statistics
    -> WHERE table_name = 'd_songs';
+------------+------------+
| INDEX_NAME | NON_UNIQUE |
+------------+------------+
| PRIMARY    |          0 |
+------------+------------+
1 row in set (0.01 sec)
```

```
mysql>
mysql> -- 6. SELECT indexes for d_events
mysql> SELECT index_name, table_name, non_unique
    -> FROM information_schema.statistics
    -> WHERE table_name = 'd_events';
Empty set (0.00 sec)

mysql>
mysql> -- 7. Create a synonym called dj_tracks
mysql> CREATE OR REPLACE VIEW dj_tracks AS
    -> SELECT * FROM d_track_listings;
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> -- 8. Create a function-based index for last_name in d_partners
mysql> CREATE INDEX idx_lower_last_name ON d_partners ((LOWER(last_name)));
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
mysql> SELECT * FROM d_partners WHERE LOWER(last_name) = 'smith';
+------------+-----------+
| partner_id | last_name |
+------------+-----------+
|          1 | Smith     |
+------------+-----------+
1 row in set (0.00 sec)

mysql>
mysql> -- 9. Create a synonym for d_track_listings and confirm
mysql> CREATE OR REPLACE VIEW syn_d_track_listings AS
    -> SELECT * FROM d_track_listings;
Query OK, 0 rows affected (0.01 sec)
```