- Descending indexes are supported for ordinary (nongenerated) and generated columns (both VIRTUAL and STORED).
- DISTINCT can use any index containing matching columns, including descending key parts.
- Indexes that have descending key parts are not used for MIN()/MAX() optimization of queries that invoke aggregate functions but do not have a GROUP BY clause.
- Descending indexes are supported for BTREE but not HASH indexes. Descending indexes are not supported for FULLTEXT or SPATIAL indexes.

  Explicitly specified ASC and DESC designators for HASH, FULLTEXT, and SPATIAL indexes results in an error.

You can see in the **Extra** column of the output of EXPLAIN that the optimizer is able to use a descending index, as shown here:

```
mysql> CREATE TABLE t1 (
    -> a INT,
    -> b INT,
    -> INDEX a_desc_b_asc (a DESC, b ASC)
    -> );

mysql> EXPLAIN SELECT * FROM t1 ORDER BY a ASC\G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: t1
   partitions: NULL
         type: index
possible_keys: NULL
          key: a_desc_b_asc
      key_len: 10
          ref: NULL
         rows: 1
     filtered: 100.00
        Extra: Backward index scan; Using index
```

In EXPLAIN FORMAT=TREE output, use of a descending index is indicated by the addition of (reverse) following the name of the index, like this:

```
mysql> EXPLAIN FORMAT=TREE SELECT * FROM t1 ORDER BY a ASC\G
*************************** 1. row ***************************
EXPLAIN: -> Index scan on t1 using a_desc_b_asc (reverse)  (cost=0.35 rows=1)
```

See also EXPLAIN Extra Information.

8.3.14 Indexed Lookups from TIMESTAMP Columns

Temporal values are stored in TIMESTAMP columns as UTC values, and values inserted into and retrieved from TIMESTAMP columns are converted between the session time zone and UTC. (This is the same type of conversion performed by the CONVERT_TZ() function. If the session time zone is UTC, there is effectively no time zone conversion.)

Due to conventions for local time zone changes such as Daylight Saving Time (DST), conversions between UTC and non-UTC time zones are not one-to-one in both directions. UTC values that are distinct may not be distinct in another time zone. The following example shows distinct UTC values that become identical in a non-UTC time zone:

```
mysql> CREATE TABLE tstable (ts TIMESTAMP);
mysql> SET time_zone = 'UTC'; -- insert UTC values
mysql> INSERT INTO tstable VALUES
```