

# **ADBMS Lab**

**Submitted by: Submitted to:**

**NAME : Rahul Gusain Dr.Ankit Khare**

**SAP: 500084143**

**ROLL NO.: R214220900**

**BATCH: 1 DevOps**

**LAB 7**

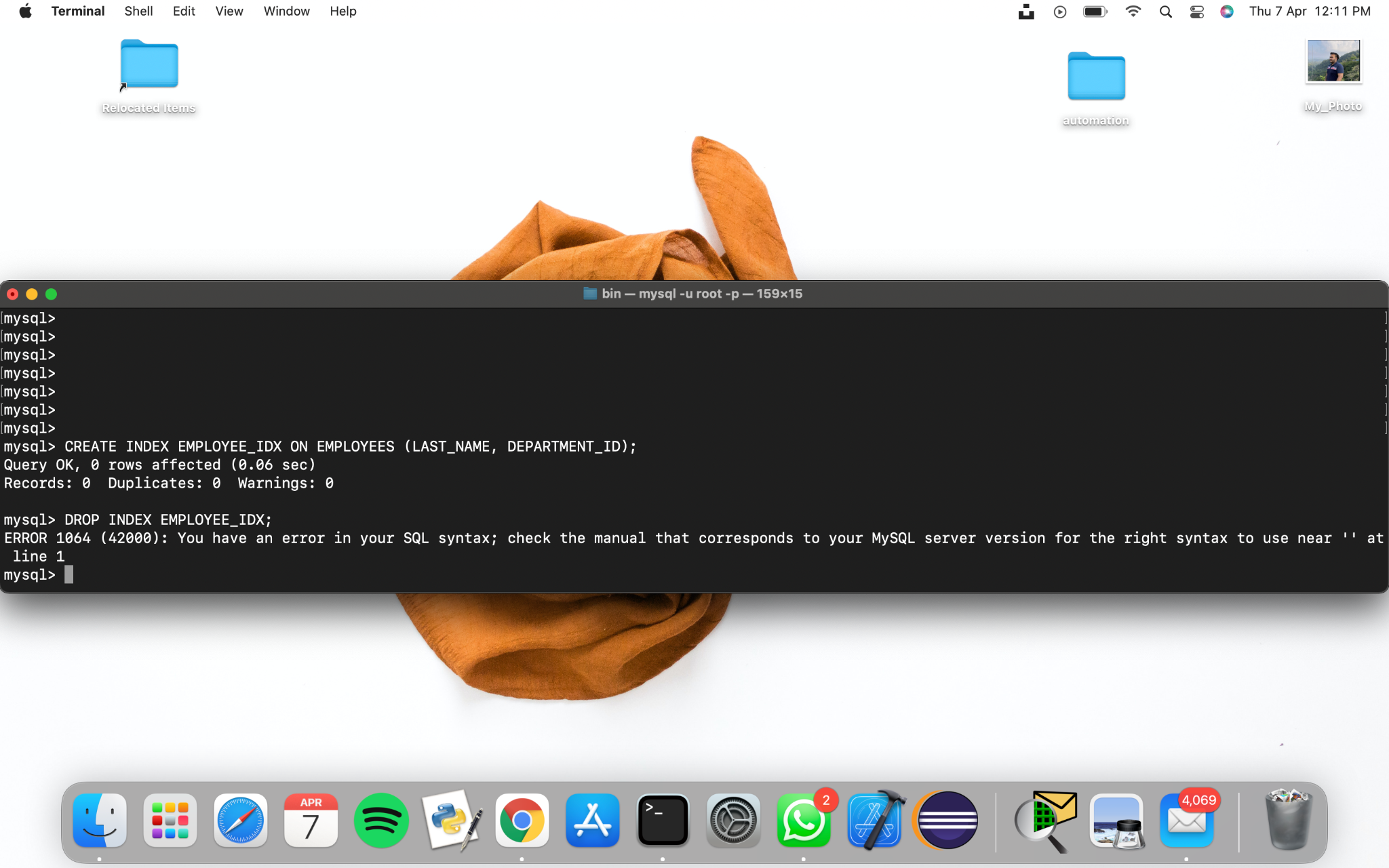
**EXPERIMENT-7**

**Title: 7. To understand the concepts of Index.   
Objective:** Students will be able to implement the concept of index.

**1. Execute the following index related queries:**

1) Create an index of name employee\_idx on EMPLOYEES with column Last\_Name, Department\_id

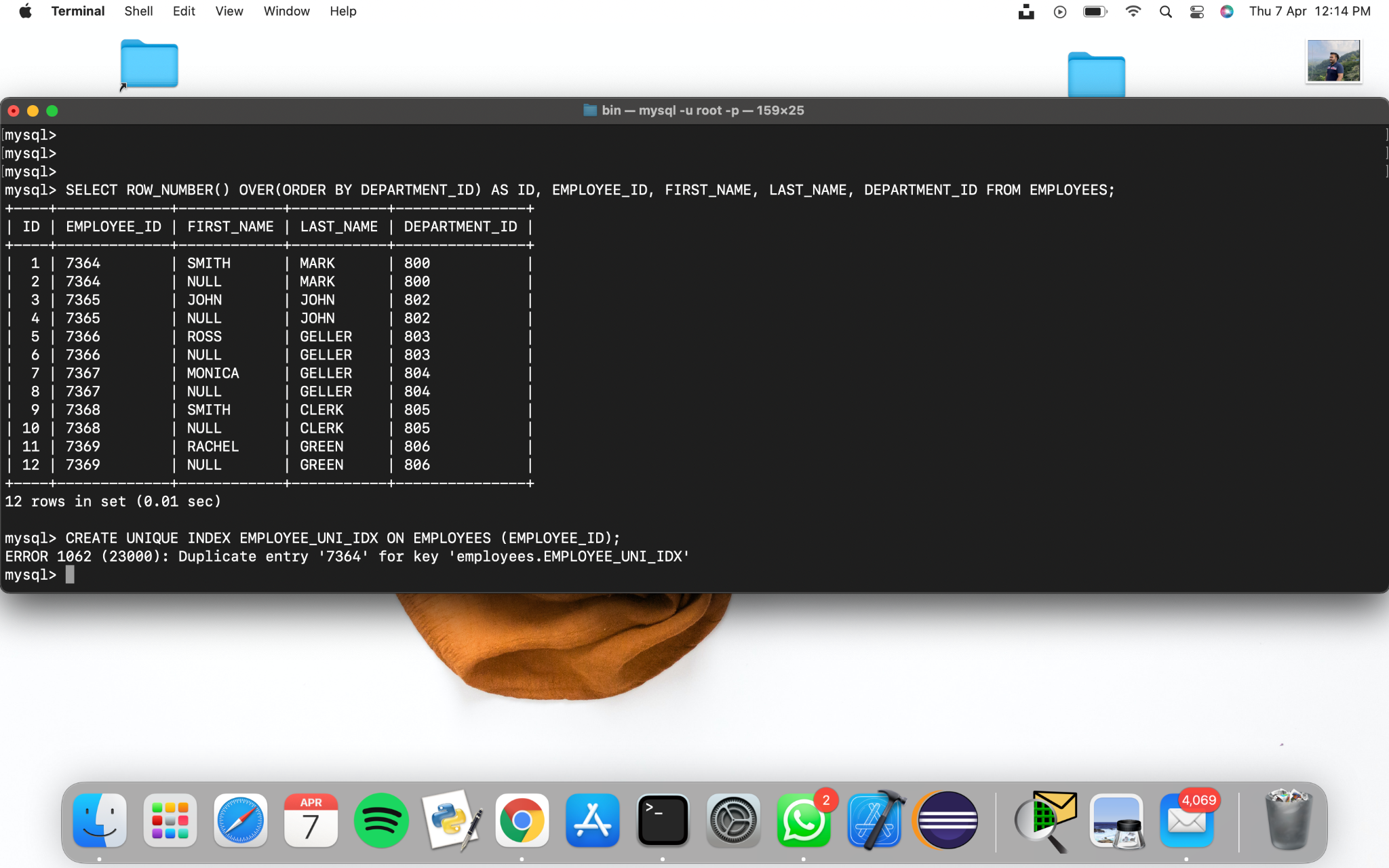
CREATE INDEX EMPLOYEE\_IDX ON EMPLOYEES (LAST\_NAME, DEPARTMENT\_ID);



**2) Find the ROWID for the above table and create a unique index on employee\_id column of the EMPLOYEES.**

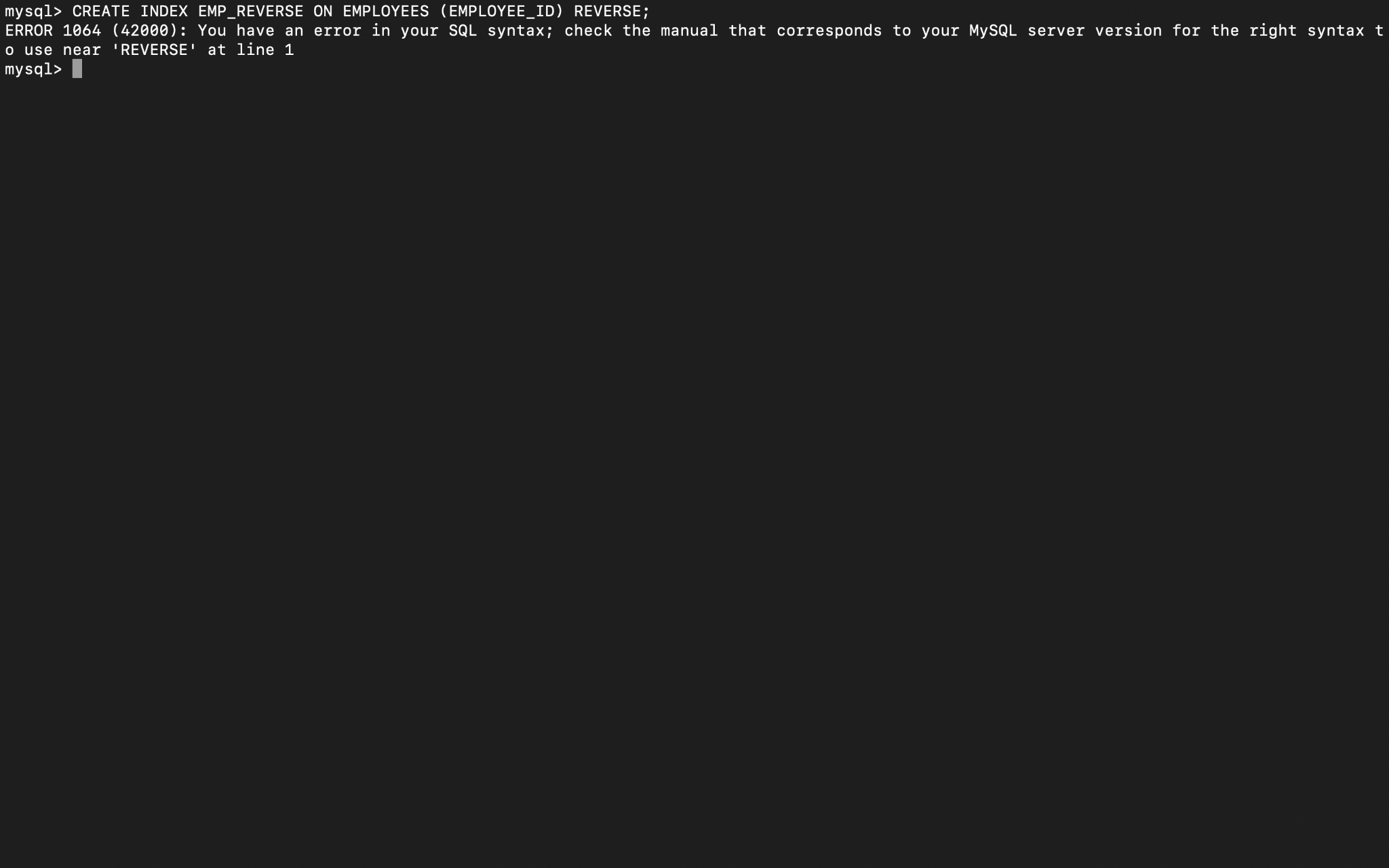
SELECT ROW\_NUMBER() OVER(ORDER BY DEPARTMENT\_ID) AS ID, EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, DEPARTMENT\_ID FROM EMPLOYEES;

CREATE UNIQUE INDEX EMPLOYEE\_UNI\_IDX ON EMPLOYEES (EMPLOYEE\_ID);



**3) Create a reverse index on employee\_id column of the EMPLOYEES.**

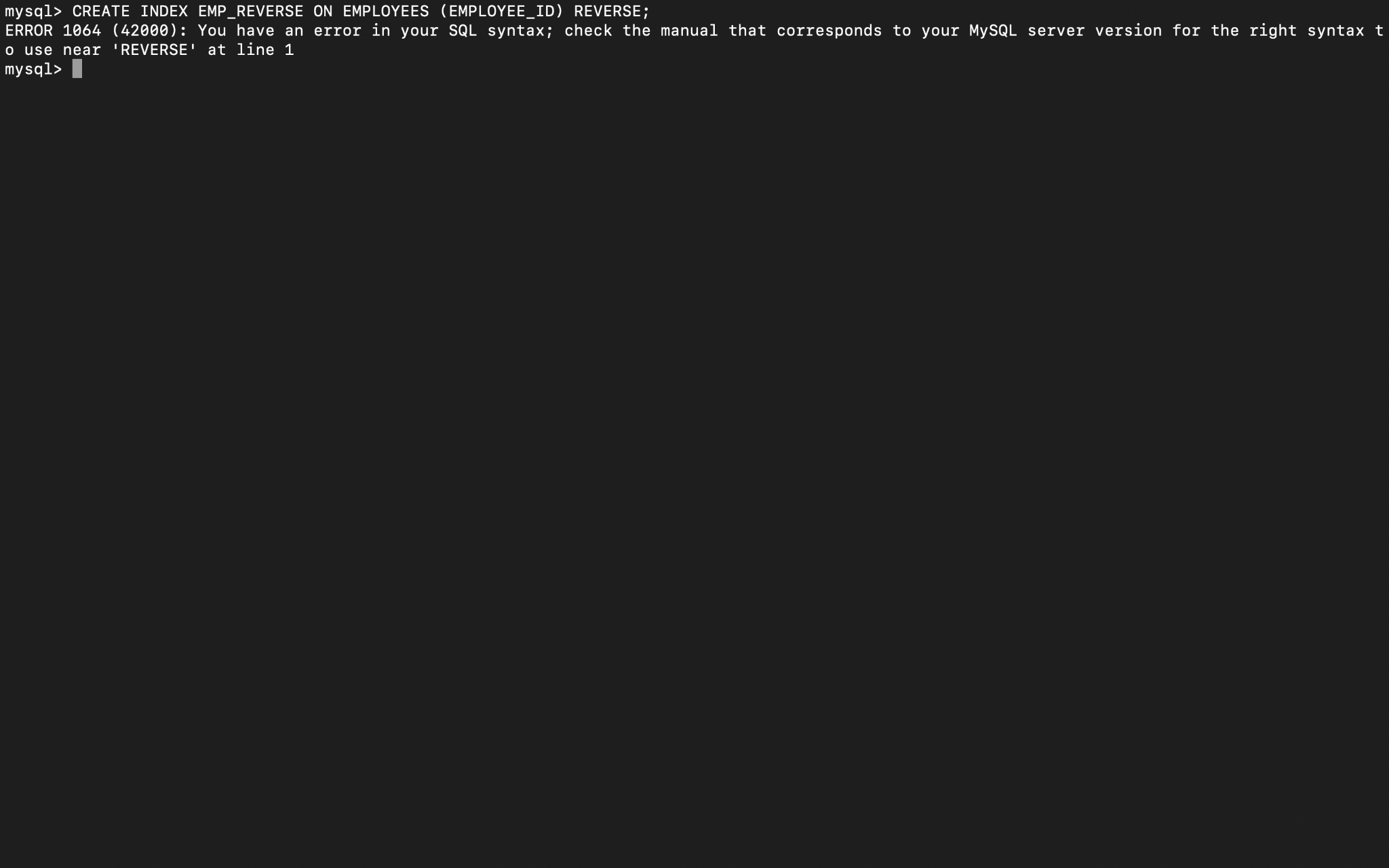
CREATE INDEX EMP\_REVERSE ON EMPLOYEES (EMPLOYEE\_ID) REVERSE;



**4) Create a unique and composite index on employee\_id and check whether there is duplicity of tuples or not.**

CREATE UNIQUE INDEX EMPLOYEE\_UNI\_COMP ON EMPLOYEES (EMPLOYEE\_ID, DEPARTMENT\_ID);

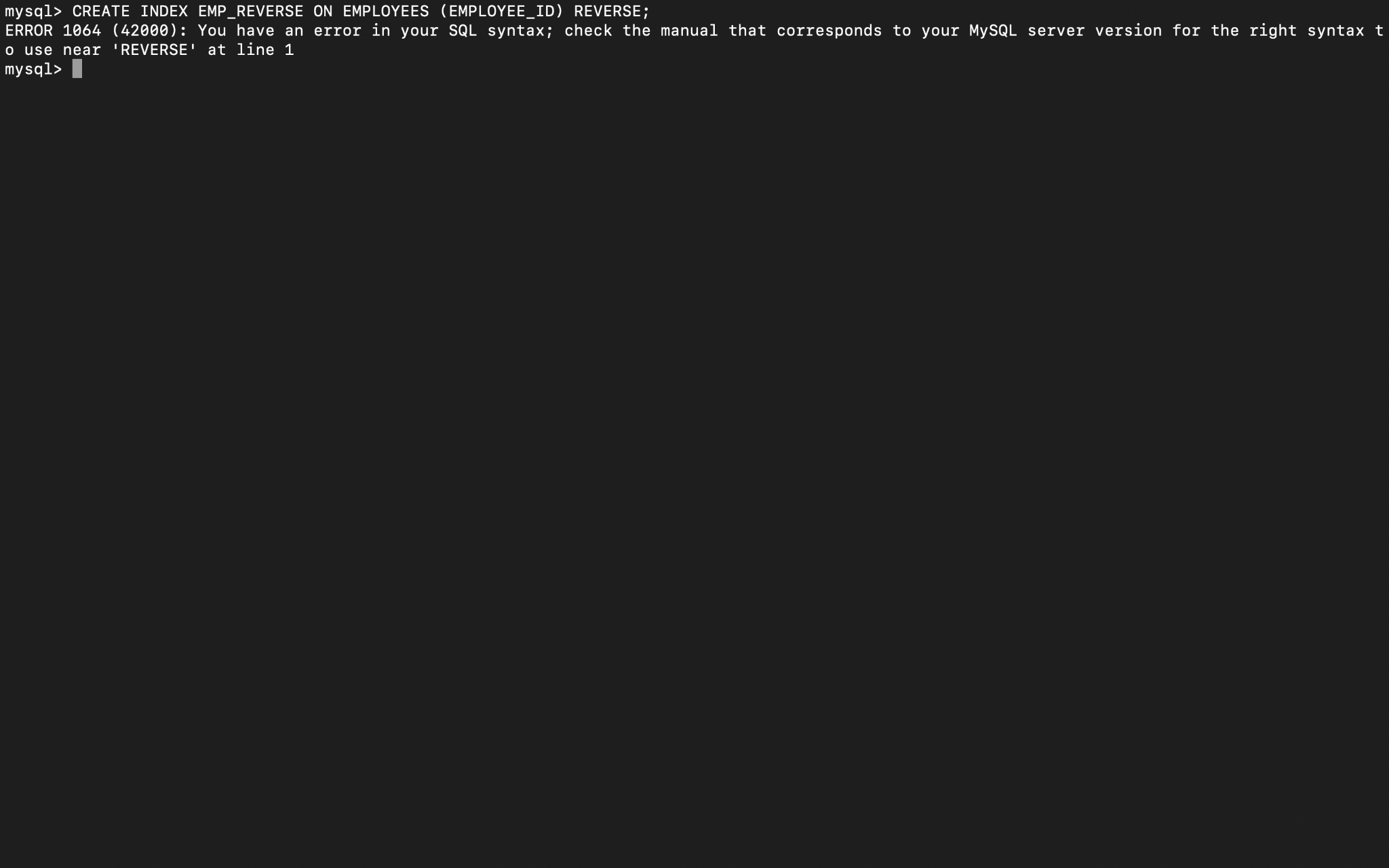
SELECT \* FROM EMPLOYEES WHERE EMPLOYEE\_ID = 'E1' AND DEPARTMENT\_ID = 'DP4';



**5) Create Function-based indexes defined on the SQL functions UPPER(column\_name) or LOWER(column\_name) to facilitate case-insensitive searches(on column Last\_Name).**

CREATE INDEX LAST\_NAME\_IDX ON EMPLOYEES (UPPER(LAST\_NAME));

CREATE INDEX LAST\_NAME\_IDX2 ON EMPLOYEES (lower(LAST\_NAME));



**6) Drop the function based index on column Last\_Name.**

DROP INDEX LAST\_NAME\_IDX;

DROP INDEX LAST\_NAME\_IDX2;

