

Possiamo simulare la connessione di 2 utenti che concorrono all'uso del database p271_prenotVoli (documento MySQL CLC) aprendo 2 connessioni a **MySQL CLC** e attivando in entrambe le Transazioni.

ESEMPIO 1 :

Nella tabella che segue sono riportati i comandi SQL che vanno sottomessi nell'ordine indicato dal progressivo in modo da rilevare gli effetti delle Transazioni:

<p>----- 2)</p> <p>USE p271_prenotVoli;</p> <p>START TRANSACTION;</p> <p>INSERT INTO prenotazioni (codVolo, dataPart, passeggero) VALUES ("XZ133" , "2018-09-30" , "Topolino");</p> <p>SELECT * , NOW() FROM prenotazioni WHERE codVolo = "XZ133";</p>	<p>----- 1)</p> <p>USE p271_prenotVoli;</p> <p>START TRANSACTION;</p> <p>INSERT INTO prenotazioni (codVolo, dataPart, passeggero) VALUES ("XZ133" , "2018-09-30" , "Pippo");</p> <p>SELECT * , NOW() FROM prenotazioni WHERE codVolo = "XZ133";</p>
<p>----- 4)</p> <p>COMMIT;</p> <p>SELECT * , NOW() FROM prenotazioni WHERE codVolo = "XZ133";</p>	<p>----- 3)</p> <p>ROLLBACK;</p> <p>SELECT * , NOW() FROM prenotazioni WHERE codVolo = "XZ133";</p>

Entrambi gli utenti vedono solo i dati consolidati e quelli che ha inserito, pertanto dopo i passi **1)** e **2)** **l'utente di sinistra** vede la prenotazione di **Topolino** mentre **l'utente di destra** vede la prenotazione di **Pippo**. Alla chiusura delle transazioni (risp.con **COMMIT** e **ROLLBACK**) vengono consolidati gli aggiornamenti pertanto entrambi gli utenti vedranno solo la prenotazione di **Topolino**

```

mysql>
mysql> USE p271_prenotVoli;
Database changed
mysql>
mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql> INSERT INTO prenotazioni
-> (codVolo, dataPart, passeggero)
-> VALUES ( "XZ133" , "2018-09-30" , "Topolino" );
Query OK, 1 row affected (0.01 sec)

mysql>
mysql> SELECT * , NOW()
-> FROM prenotazioni
-> WHERE codVolo = "XZ133";
+----+-----+-----+-----+-----+
| ID | codVolo | dataPart | Passeggero | NOW() |
+----+-----+-----+-----+-----+
| 1  | XZ133  | 2018-09-30 | Bianchi Filippo | 2024-03-14 18:35:24 |
| 2  | XZ133  | 2018-09-30 | Rossi Eugenio | 2024-03-14 18:35:24 |
| 3  | XZ133  | 2018-09-30 | Batini Franca | 2024-03-14 18:35:24 |
| 21 | XZ133  | 2018-09-30 | Topolino | 2024-03-14 18:35:24 |
+----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> COMMIT;
Query OK, 0 rows affected (0.02 sec)

mysql> SELECT * , NOW()
-> FROM prenotazioni
-> WHERE codVolo = "XZ133";
+----+-----+-----+-----+-----+
| ID | codVolo | dataPart | Passeggero | NOW() |
+----+-----+-----+-----+-----+
| 1  | XZ133  | 2018-09-30 | Bianchi Filippo | 2024-03-14 18:37:37 |
| 2  | XZ133  | 2018-09-30 | Rossi Eugenio | 2024-03-14 18:37:37 |
| 3  | XZ133  | 2018-09-30 | Batini Franca | 2024-03-14 18:37:37 |
| 21 | XZ133  | 2018-09-30 | Topolino | 2024-03-14 18:37:37 |
+----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>

```

```

mysql>
mysql> USE p271_prenotVoli;
Database changed
mysql>
mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql> INSERT INTO prenotazioni
-> (codVolo, dataPart, passeggero)
-> VALUES ( "XZ133" , "2018-09-30" , "Pippo" );
Query OK, 1 row affected (0.02 sec)

mysql>
mysql> SELECT * , NOW()
-> FROM prenotazioni
-> WHERE codVolo = "XZ133";
+----+-----+-----+-----+-----+
| ID | codVolo | dataPart | Passeggero | NOW() |
+----+-----+-----+-----+-----+
| 1  | XZ133  | 2018-09-30 | Bianchi Filippo | 2024-03-14 18:35:13 |
| 2  | XZ133  | 2018-09-30 | Rossi Eugenio | 2024-03-14 18:35:13 |
| 3  | XZ133  | 2018-09-30 | Batini Franca | 2024-03-14 18:35:13 |
| 20 | XZ133  | 2018-09-30 | Pippo | 2024-03-14 18:35:13 |
+----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> ROLLBACK;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * , NOW()
-> FROM prenotazioni
-> WHERE codVolo = "XZ133";
+----+-----+-----+-----+-----+
| ID | codVolo | dataPart | Passeggero | NOW() |
+----+-----+-----+-----+-----+
| 1  | XZ133  | 2018-09-30 | Bianchi Filippo | 2024-03-14 18:37:40 |
| 2  | XZ133  | 2018-09-30 | Rossi Eugenio | 2024-03-14 18:37:40 |
| 3  | XZ133  | 2018-09-30 | Batini Franca | 2024-03-14 18:37:40 |
| 21 | XZ133  | 2018-09-30 | Topolino | 2024-03-14 18:37:40 |
+----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>

```

ESEMPIO 2 :

Si procede agli inserimenti delle stesse prenotazioni previo controllo della disponibilità del posto (contando le prenotazioni già inserite per il volo che devono essere inferiori al numero di posti dell'aereo).

<p>----- 2)</p> <pre>START TRANSACTION; SELECT V.* , COUNT(ID) AS occupati FROM prenotazioni P RIGHT JOIN voli V USING (codVolo) WHERE codVolo = "XZ133"; INSERT INTO prenotazioni (codVolo, dataPart, passeggero) VALUES ("XZ133" , "2018-09-30" , "Topolino"); SELECT * , NOW() FROM prenotazioni WHERE codVolo = "XZ133";</pre>	<p>----- 1)</p> <pre>START TRANSACTION; SELECT V.* , COUNT(ID) AS occupati FROM prenotazioni P RIGHT JOIN voli V USING (codVolo) WHERE codVolo = "XZ133"; INSERT INTO prenotazioni (codVolo, dataPart, passeggero) VALUES ("XZ133" , "2018-09-30" , "Pippo"); SELECT * , NOW() FROM prenotazioni WHERE codVolo = "XZ133";</pre>
<p>----- 4)</p> <pre>COMMIT; SELECT * , NOW() FROM prenotazioni WHERE codVolo = "XZ133";</pre>	<p>----- 3)</p> <pre>ROLLBACK; SELECT * , NOW() FROM prenotazioni WHERE codVolo = "XZ133";</pre>

C'è un ultimo posto disponibile, ed entrambi gli utenti lo vedono come tale e procedono all'inserimento; se entrambi effettuassero il COMMIT si determinerebbe una prenotazione in overbooking per un totale di 6 posti prenotati.

The image shows two side-by-side terminal windows. The left window shows a user starting a transaction, checking the seat availability for flight XZ133 (4 seats occupied), and inserting a passenger named 'Topolino'. The right window shows another user doing the same check (4 seats occupied) and inserting a passenger named 'Pippo'. Both users then commit their transactions. The final output of the left window shows 5 rows in the 'prenotazioni' table, with the last row being 'Topolino'. The final output of the right window shows 5 rows, with the last row being 'Pippo'. Red circles highlight the 'posti' and 'occupati' columns in the initial SELECT queries of both windows, which both show a value of 4.

codVolo	aereo	posti	occupati
XZ133	PA2-Piper light aircratf	5	4

ID	codVolo	dataPart	Passeggero	NOW()
1	XZ133	2018-09-30	Bianchi Filippo	2024-03-14 19:15:24
2	XZ133	2018-09-30	Rossi Eugenio	2024-03-14 19:15:24
3	XZ133	2018-09-30	Batini Franca	2024-03-14 19:15:24
25	XZ133	2018-09-30	Topolino	2024-03-14 19:15:24

ID	codVolo	dataPart	Passeggero	NOW()
1	XZ133	2018-09-30	Bianchi Filippo	2024-03-14 19:15:07
2	XZ133	2018-09-30	Rossi Eugenio	2024-03-14 19:15:07
3	XZ133	2018-09-30	Batini Franca	2024-03-14 19:15:07
24	XZ133	2018-09-30	Pippo	2024-03-14 19:15:07

Per evitare questa anomalia (detta del *record fantasma*) il controllo sul numero dei posti disponibili va fatto bloccando per gli altri utenti l'accesso alle righe controllate.

Per ottenere questo, nella SELECT di controllo va usata l'opzione **FOR UPDATE**.

Questa tecnica può comportare il blocco per un tempo eccessivo degli altri utenti ovvero la **chiusura automatica della transazione in attesa dopo un tempo prefissato dal DBMS** (1 minuto ad esempio) con conseguente ROLLBACK automatico e segnalazione di errore

<pre>----- 2) START TRANSACTION; SELECT V.* , COUNT(ID) AS occupati FROM prenotazioni P RIGHT JOIN voli V USING (codVolo) WHERE codVolo = "XZ133" FOR UPDATE;</pre>	<pre>----- 1) START TRANSACTION; SELECT V.* , COUNT(ID) AS occupati FROM prenotazioni P RIGHT JOIN voli V USING (codVolo) WHERE codVolo = "XZ133" FOR UPDATE;</pre>
--	--

```
Prompt dei comandi - mysql -u root -p
mysql>
mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> SELECT V.* , COUNT(ID) AS occupati
-> FROM   prenotazioni P
->       RIGHT JOIN voli V      USING (codVolo)
->       WHERE codVolo = "XZ133"
-> FOR UPDATE;
```

```
Prompt dei comandi - mysql -u root -p
mysql>
mysql>
mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> SELECT V.* , COUNT(ID) AS occupati
-> FROM   prenotazioni P
->       RIGHT JOIN voli V      USING (codVolo)
->       WHERE codVolo = "XZ133"
-> FOR UPDATE;
```

codVolo	aereo	posti	occupati
XZ133	PA2-Piper light aircratf	4	4

```
1 row in set (0.03 sec)

mysql> _
```

La **transazione di sinistra** resta in attesa fino al consolidamento dell'inserimento effettuato dalla **transazione di destra** oppure fino alla ricezione del messaggio di errore:

ERROR 1205 (HY000): Lock wait timeout exceeded; try restarting transaction

```
mysql> SELECT V.* , COUNT(ID) AS occupati
-> FROM   prenotazioni P
->       RIGHT JOIN voli V      USING (codVolo)
->       WHERE codVolo = "XZ133"
-> FOR UPDATE;
ERROR 1205 (HY000): Lock wait timeout exceeded; try restarting transaction
mysql> _
```