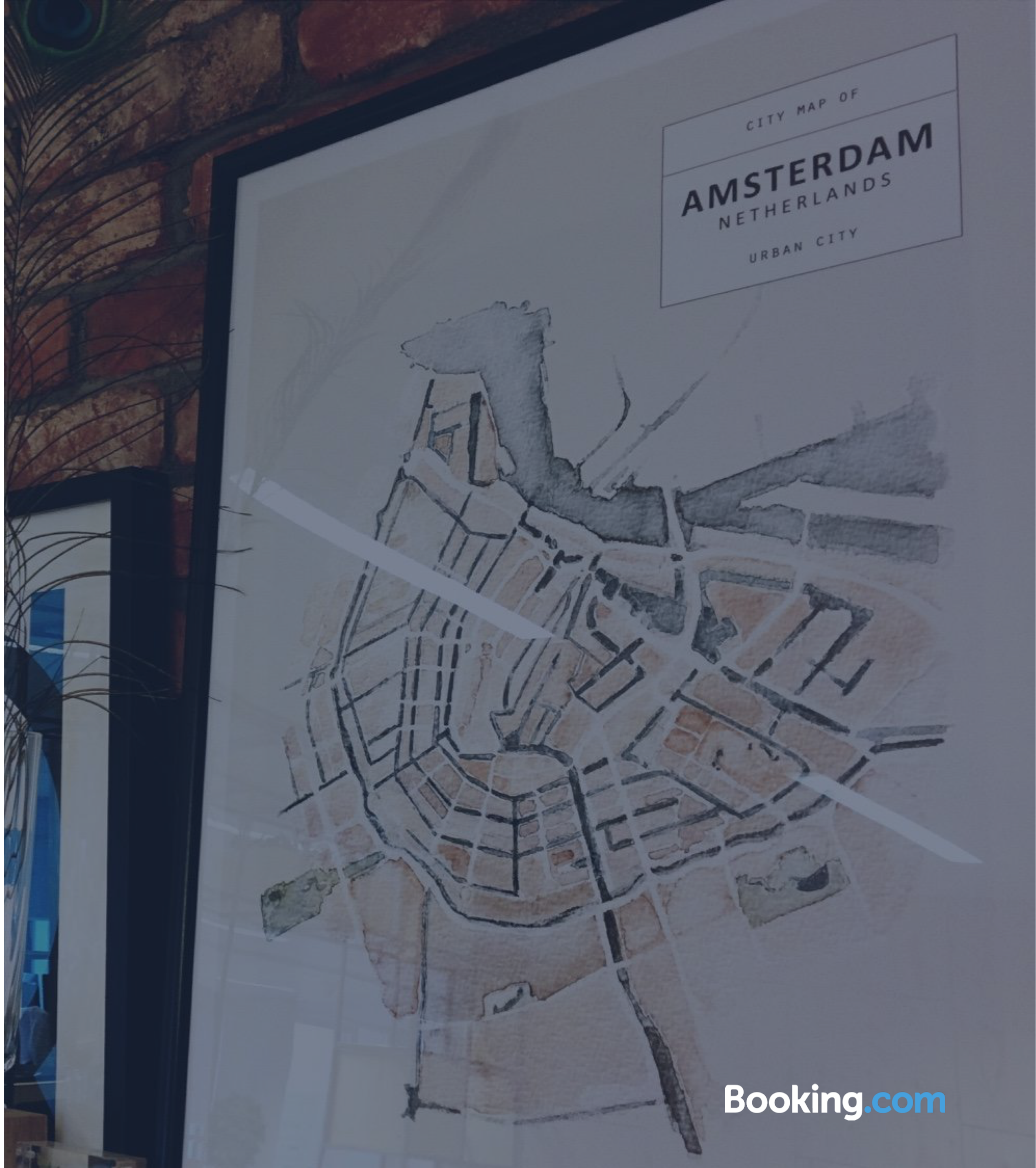
Booking.com

MySQL 8.0 - our view

Oracle Open World, 25th October 2018
Simon J Mudd <simon.mudd@booking.com>

Overview.

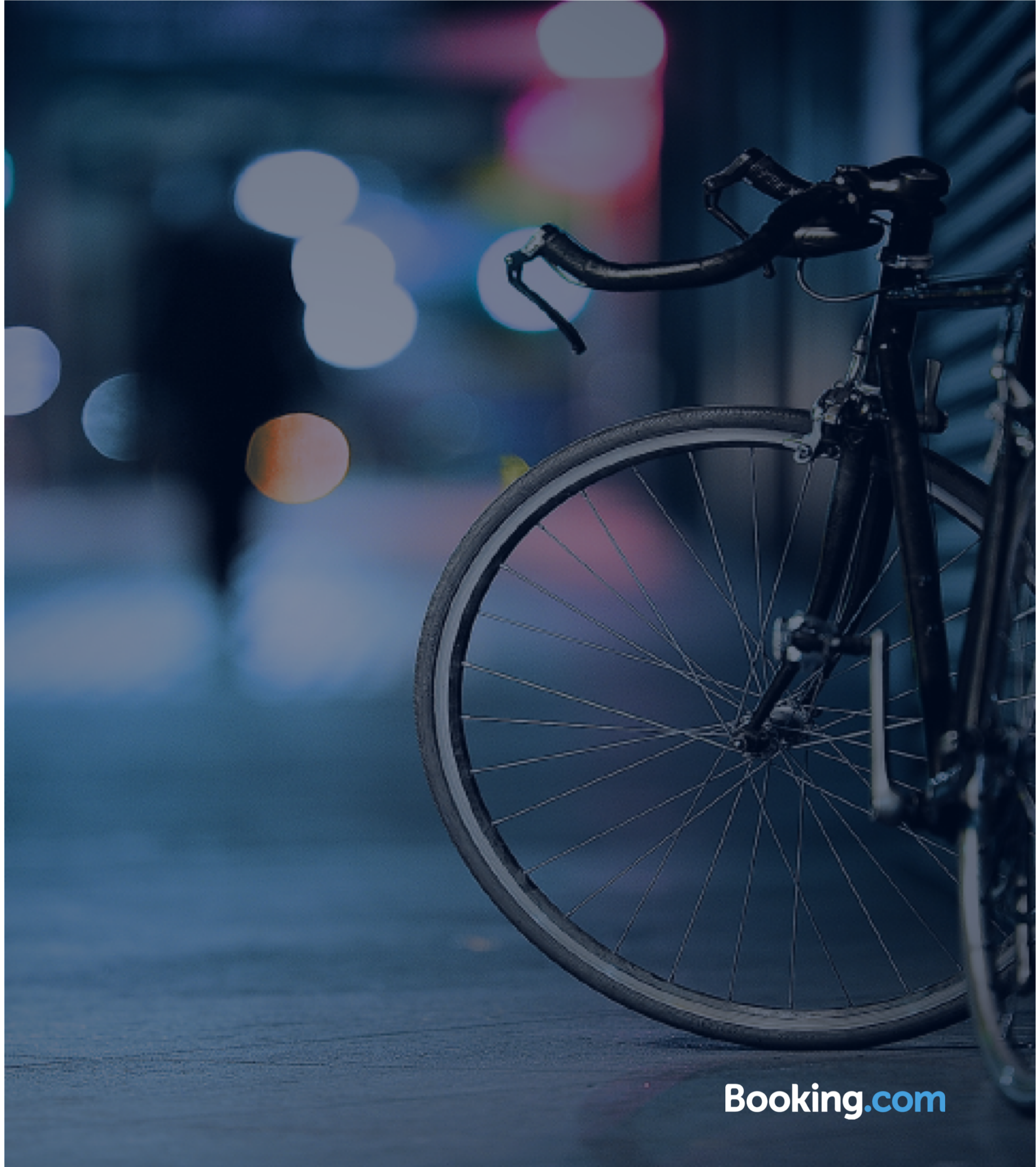
- Upgradability
- Features that interest us
  - New Data dictionary
  - TLS
  - Logging
  - Instant ALTER TABLE
  - Roles
  - Replication
  - CTEs & GIS



Booking.com

Upgradability

- We upgrade servers all the time. Many people don’t
- Despite the significant internal changes in 8.0 vs 5.7 the upgrade process is very smooth
    - Stop server, swap binaries, start server, run mysql_upgrade, restart
    - In previous DMRs, e.g. 5.7 and 5.6 we’ve often seen crashes and strange behavior but even in 8.0.0 the process was good
    - We started testing 8.0.0 (as a slave) in 09/2016, GA release 04/2018

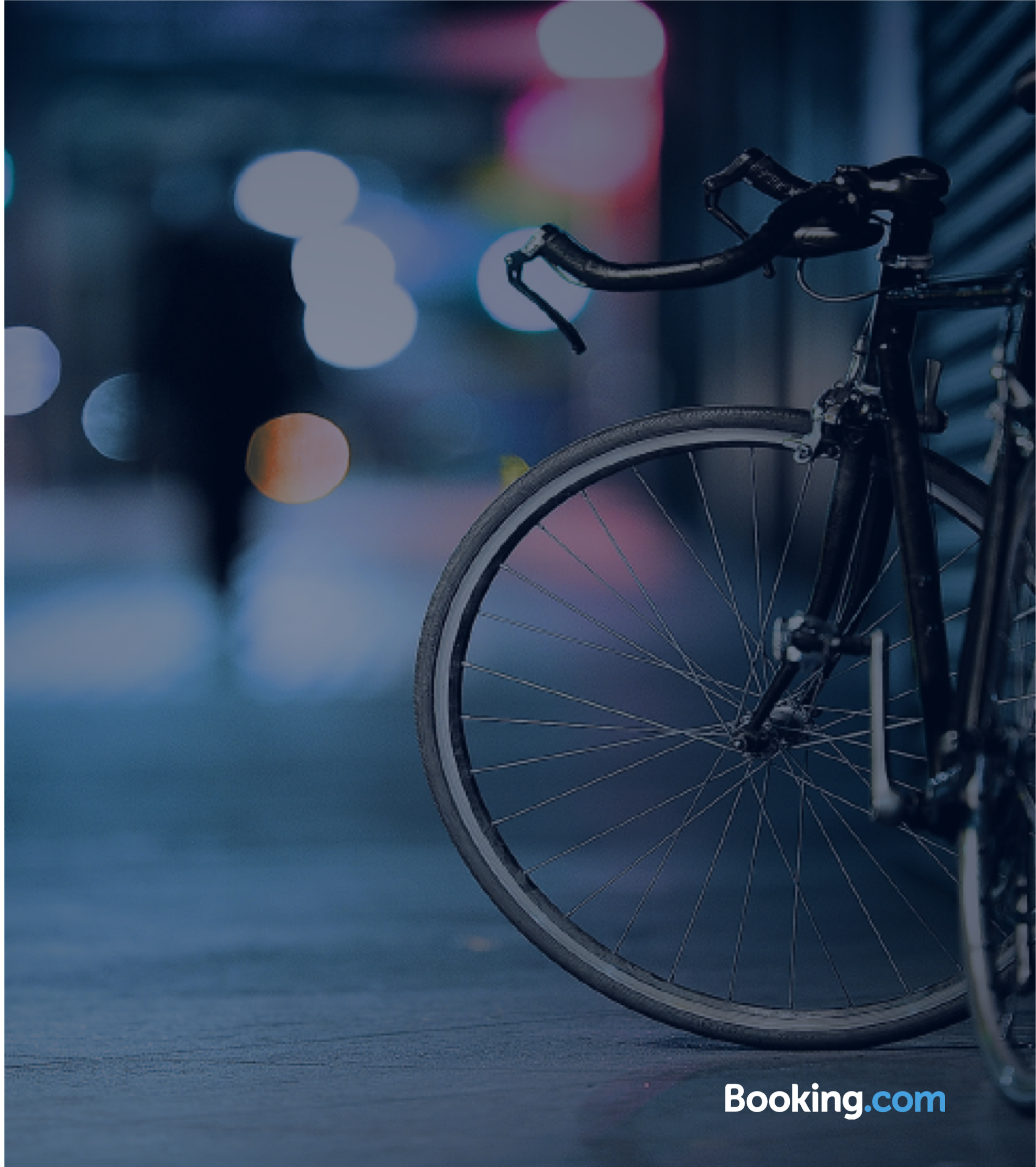


Upgradability

- Upgrade checker is welcome
  - Introduced in 8.0.4 MySQ Shell
  - Typical usage:

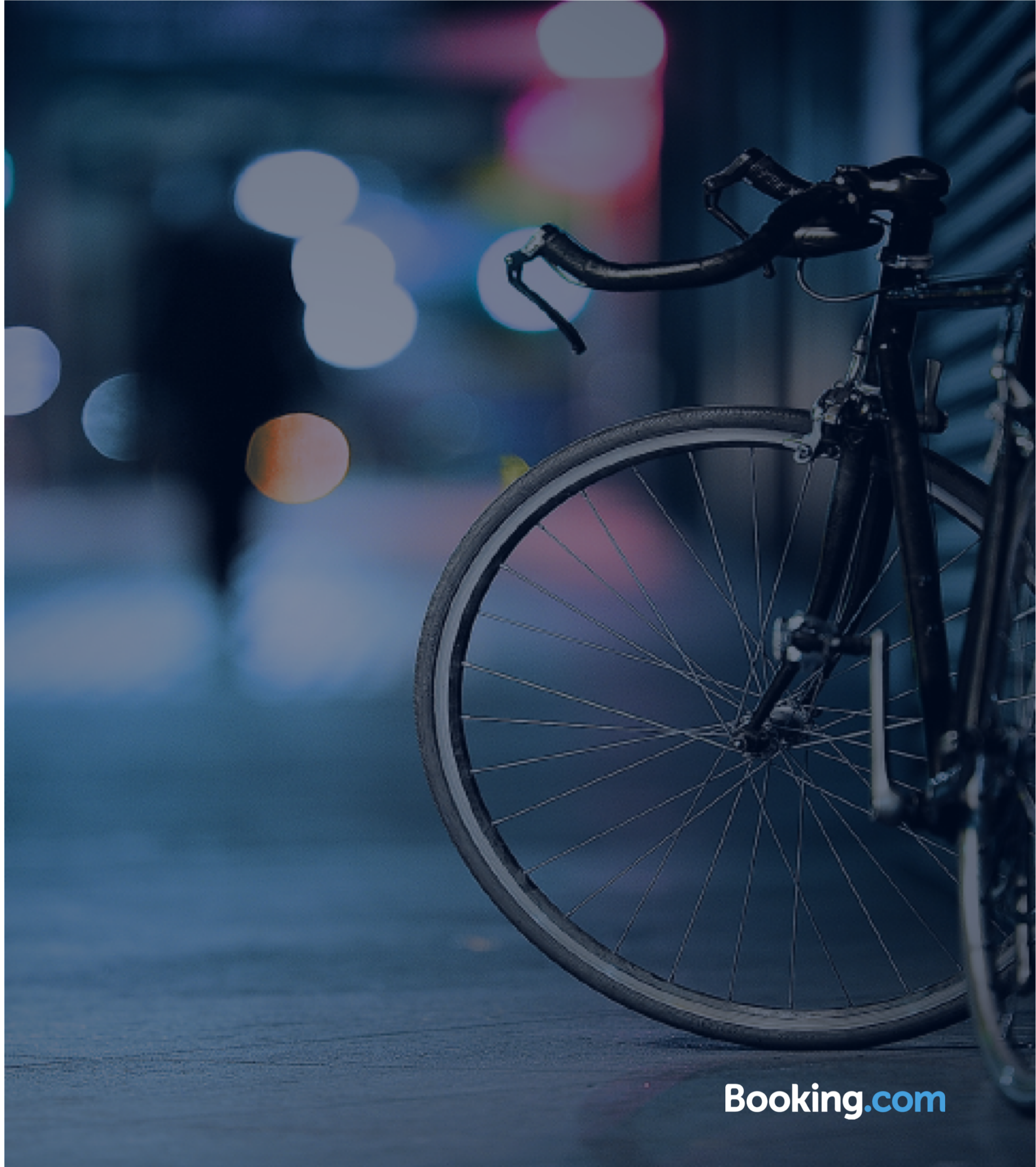
```
mysqlsh root:@localhost:3306 -e \
     "util.checkForServerUpgrade();”
```

  - Checks tables for potential problems
  - It does not seem to work properly on 8.0.13 due to some hard-coding https://bugs.mysql.com/bug.php?id=92918



Upgradability

- Even minor releases have issues
- 8.0.13 changed some logging variables which once you’re in a GA version really should be avoided as old name no longer supported
- The upgrade checker could catch this and see if you’ve configured the old variables



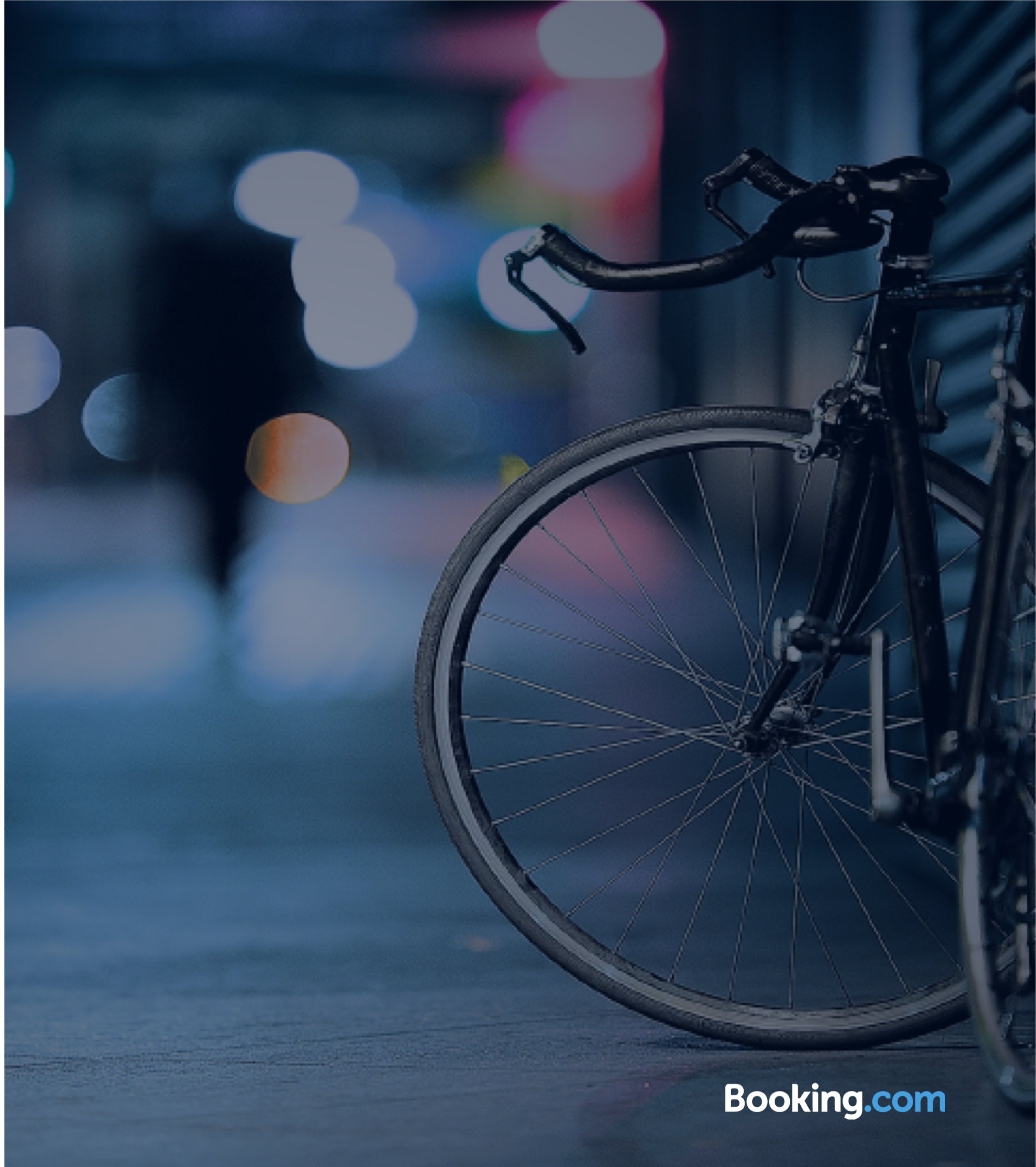
Upgradability

- New ideas to help further
  - Replication SQL checker

```
Last_SQL_Error: Error 'You have an error in your SQL
syntax; check the manual that corresponds to your MySQL
server version for the right syntax to use near 'groups,
col7) VALUES (CONNECTION_ID(), USER(), 'val3', 'val4' at
line 2' on query. Default database: 'mydb'. Query:
'INSERT INTO MyTable (connection_id, connection_user,
col3, col4, col5, groups, col7) VALUES (CONNECTION_ID(),
USER(), 'val3', 'val4', 'val5', 'val6', NULL)’
```

  - SQL checker

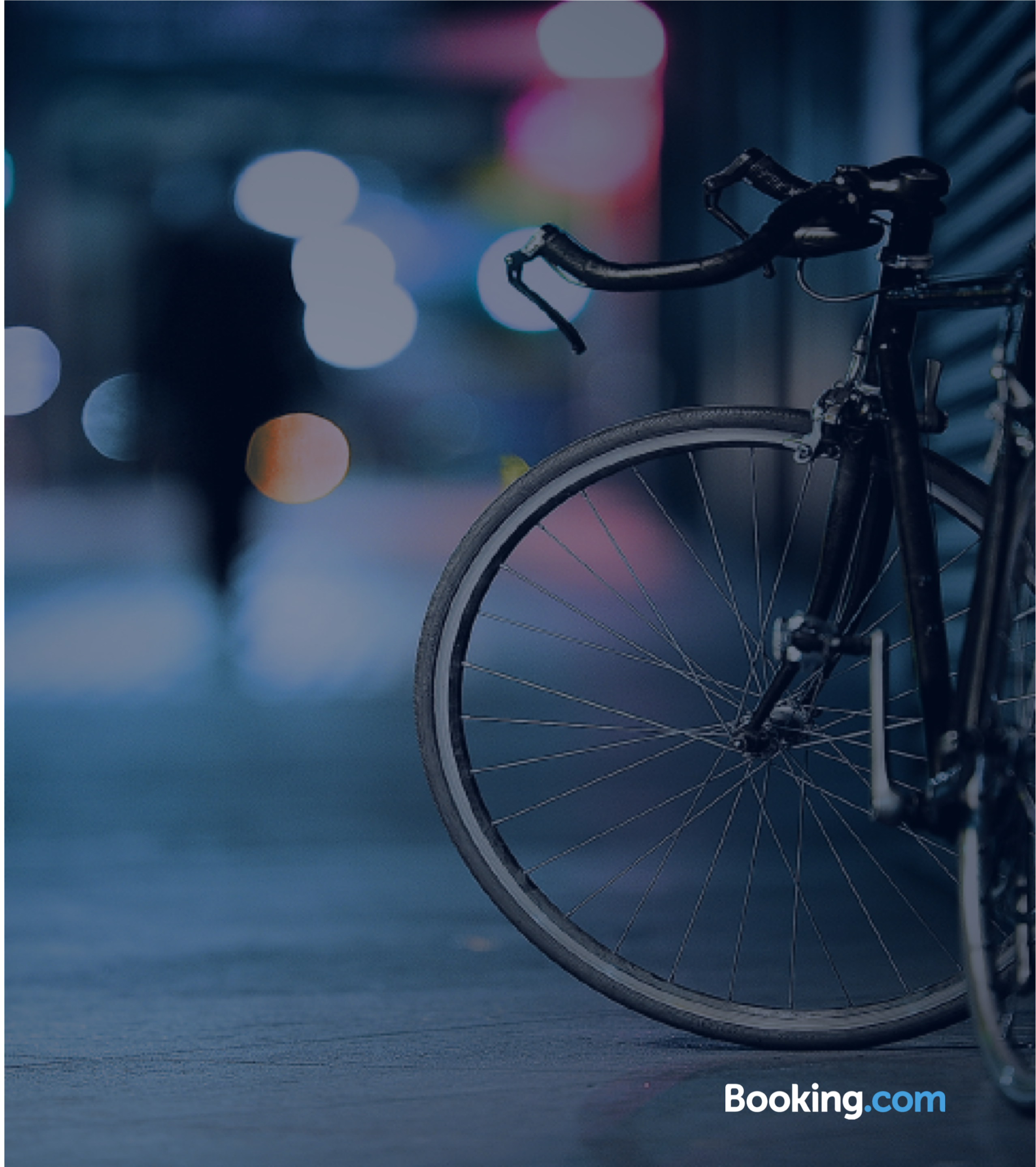
- These tools may not be necessary, but they do provide a way for users to be more comfortable that they won’t be affected by upgrading the server



Configuration changes

Few configuration changes:

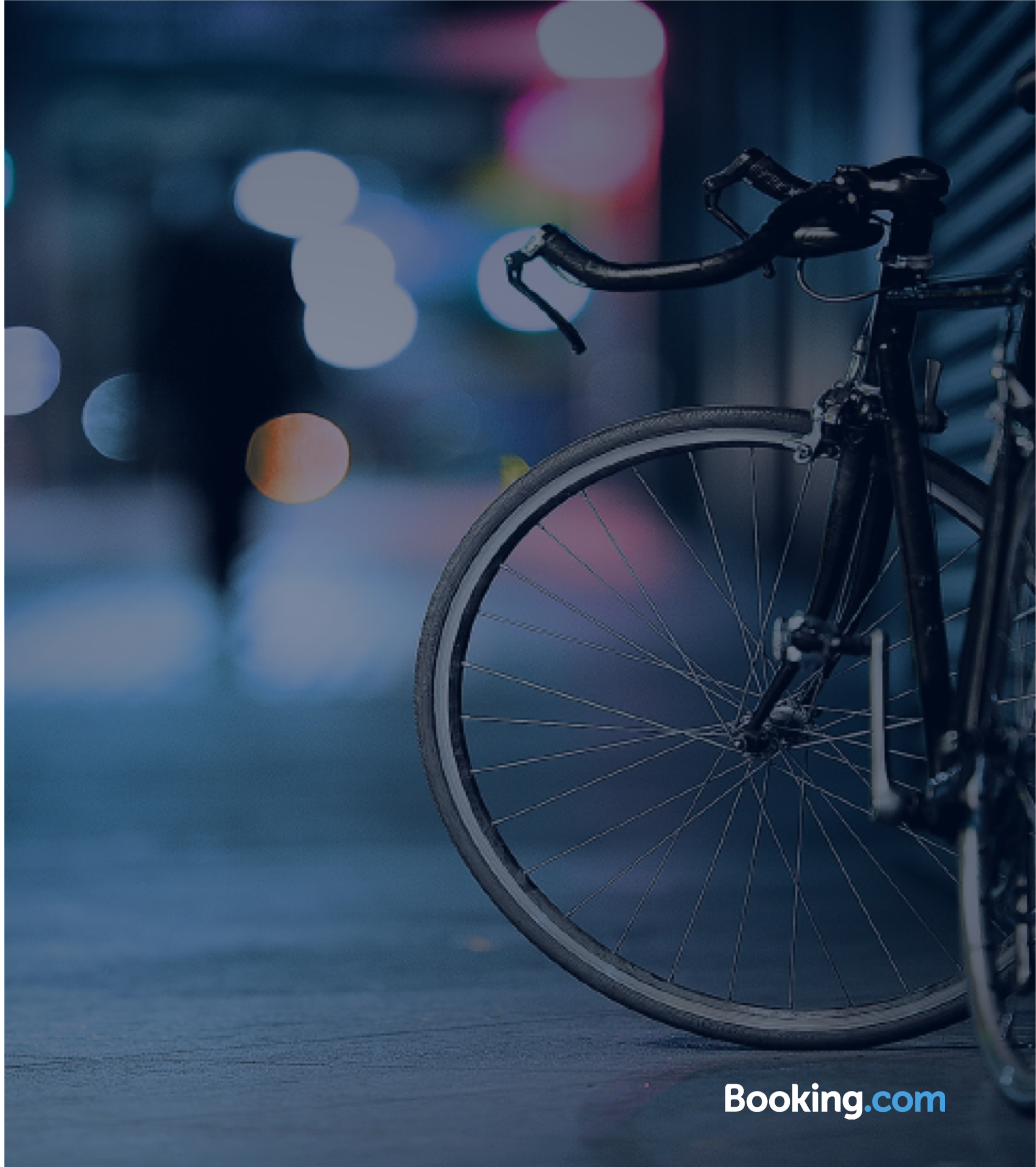
- binlog_row_metadata = full
- Allows more meta data to be sent in binlogs which is good for binlog processing (for feeding into external systems)
- Binlog_expire_seconds instead of binlog_expire_days
- default_authentication_plugin=mysql_native_password as we still run in many cases upstream 5.7 masters, and while we upgrade connectors
- default_collation_for_utf8mb4 = 'utf8mb4_general_ci' for 5.7 -> 8.0 replication compatibility



Configuration changes

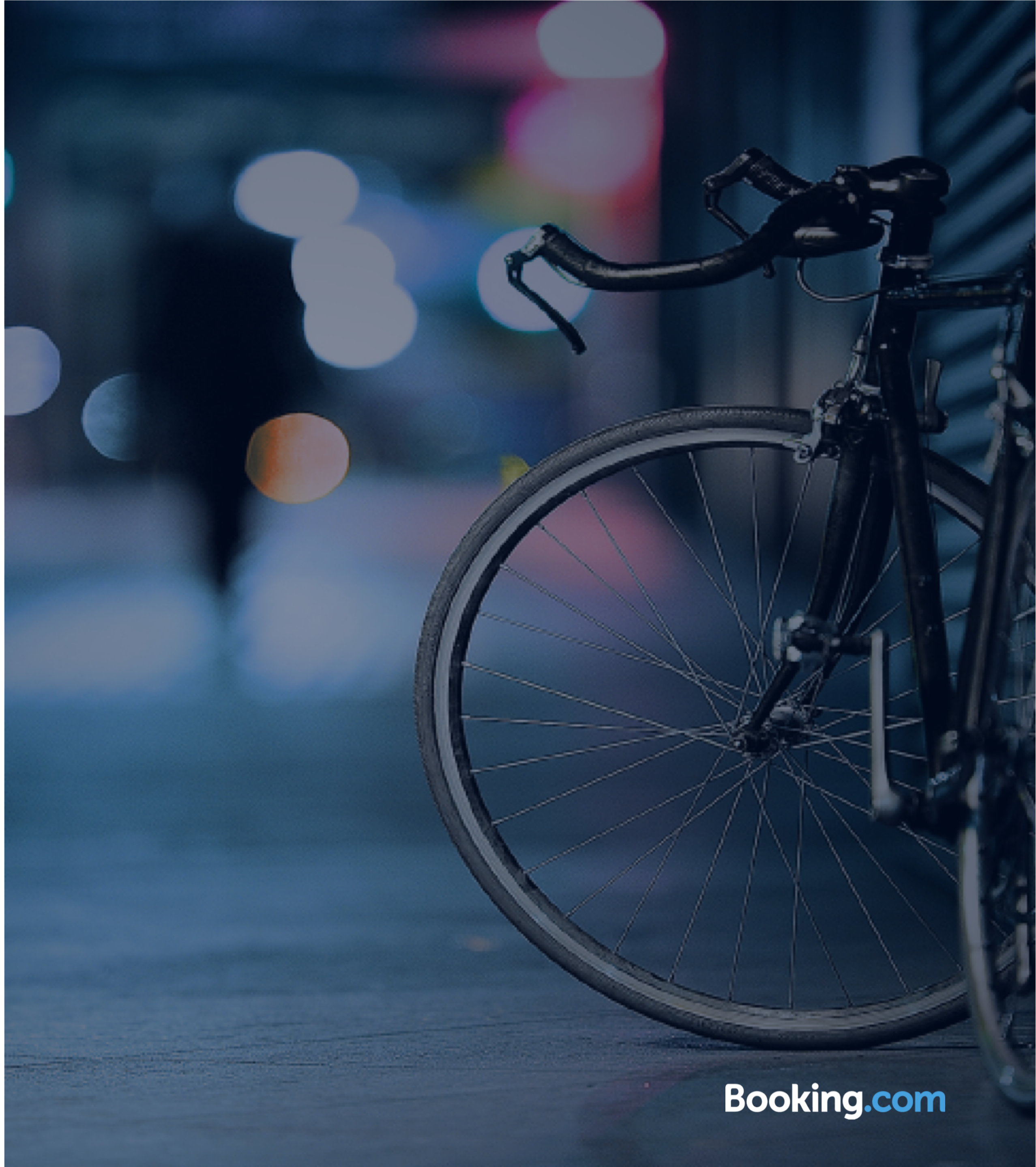
Few configuration changes (2):

- Remove query cache_* settings
- loose_sql_require_primary_key = ON



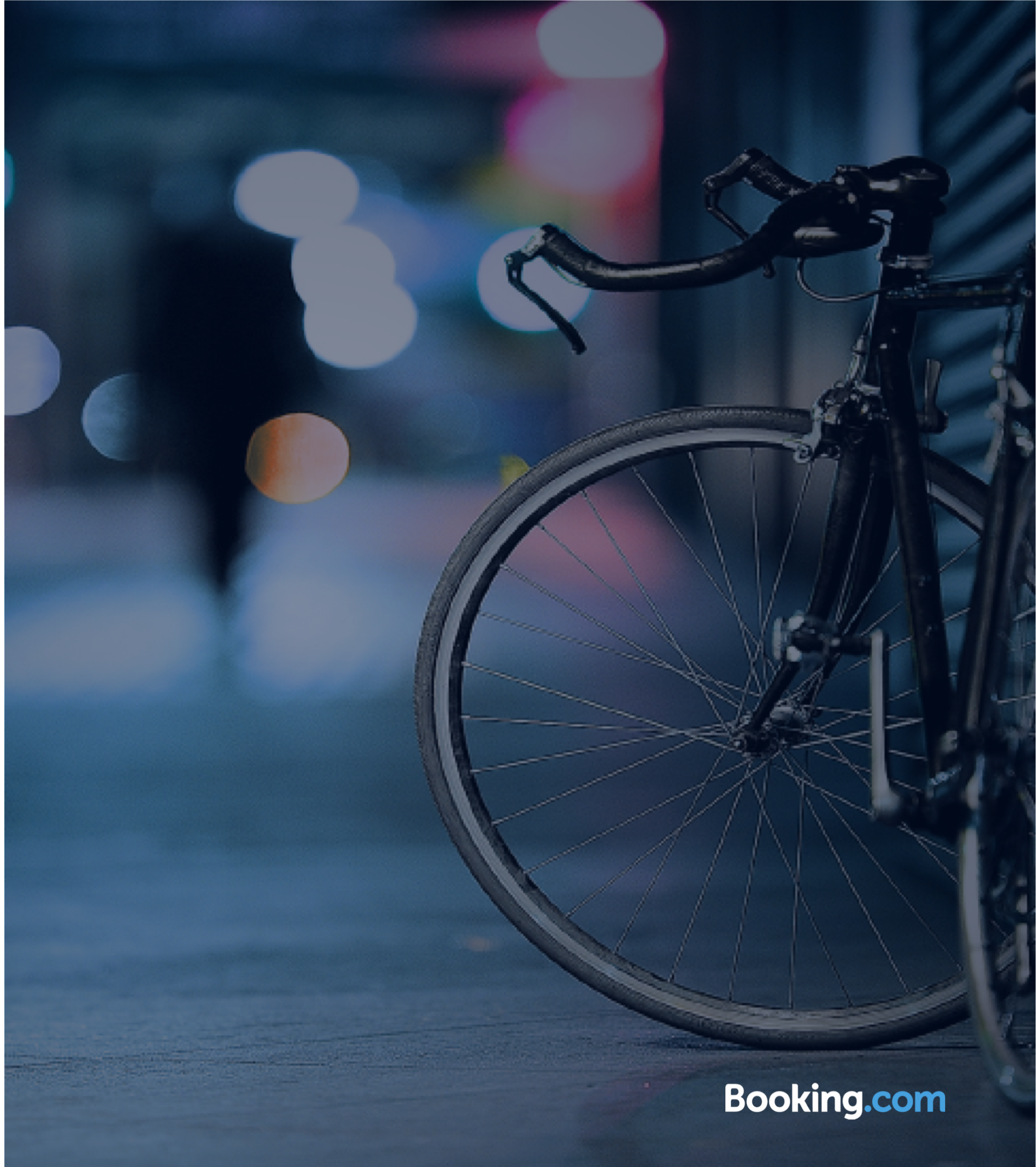
New Data Dictionary

- In theory ➔ "not interesting"
  - holds internal server metadata
- In practice ➔ yes it is
  - we've seen crashes and inconsistencies which would cause problems
  - atomic DDL is important as we do this very frequently
  - Handling large numbers of tables has always performed very slowly and for us required – so change is good
  - Storage of table metadata inside the .ibd files is good as easier to move tables and also verify internal structures



TLS

- YaSSL library used in 5.7 (from WolfSSL company) no longer updated
- 8.0 supports OpenSSL and WolfSSL
- MySQL community version now built with OpenSSL rather than YaSSL in 5.7
- OpenSSL library makes better use of new hardware features like AVX2 so faster
- OpenSSL supports TLSv1.2, YaSSL doesn’t
- More MySQL status vars updated with OpenSSL
- MySQL doesn’t yet support TLSv1.3 (provided by OpenSSL, WolfSSL)
- For us: improved performance and easier to use a better library



Logging

Logging has traditionally been quite limited in what was available

- 8.0 provides the new dragnet logging that allows for filtering and modification of log entries

```
INSTALL COMPONENT 'file://component_log_filter_dragnet’;
SET GLOBAL log_error_services='log_filter_dragnet; log_sink_internal';
SET GLOBAL dragnet.log_error_filter_rules='IF err_code == 10927 or err_code == 10914 or err_code
== 10926 then drop.’;
```

Logging

Allows you to filter out things like:

```
2018-04-23T01:31:07.379321Z 668382 [Note] [MY-010927] [Server] Access denied for user
'user'@'host'. Account is locked.
2018-04-23T12:05:26.849775Z 11286 [Note] [MY-010914] [Server] Aborted connection 11286 to db:
'unconnected' user: 'user' host: 'host' (Got an error reading communication packets).
2018-04-24T07:30:23.939494Z 616126 [Note] [MY-010926] [Server] Access denied for user
'user'@'host' (using password: NO)
```

- the structured logging, even in text format is easier to parse
- can produce JSON output which is good for forwarding on to external systems for analysis

Logging

8.0.13 allows a simpler filter:

- log_error_suppression_list

The filtering language could possibly be made a bit more complex

- Everything currently has to go in one line. That’s ugly
- No “IN clause” if you want to group a number of errors together

Bootstrap configuration still unclear to me

- https://bugs.mysql.com/90579

P_S metrics of what dragnet logging does (e.g. dropped messages) would be useful

Instant ALTER TABLE

DDLs have notoriously been hard in MySQL

- Instant ADD COLUMN is most useful and saves time
- Hoping that **other variants** of this may appear over time now the data dictionary is transactional

Wanted: EXPLAIN ALTER TABLE

- With many changes over versions of the behavior of ALTER TABLE it's important to be able to find out if the change will be instant or not. There's no way to do this now.
- The suggested ALGORITHM=INSTANT does not allow you to verify before making the change. With a replication chain of several servers of perhaps different versions it may be necessary to check that the "instant" behavior is true on all servers before attempting to actually use the ALGORITHM=INSTANT option.
  - https://bugs.mysql.com/34354

Roles

Roles in MySQL 8.0 look good.

- Better and more consistent management of privileges
- Especially useful if you have a large number of users with similar rights
  - E.g. “readers” (of a complete database or set of tables)

How are they used?

- CREATE ROLE <role_name>;
- GRANT <rights> TO <role_name>; -- rights can be an existing role
- GRANT <role_name> TO ‘user1’@’%’; -- push the right to a real user
- GRANT <role_name> TO ‘user2’@’%’;

Roles

Accounts with multiple roles

- For human access to prevent escalating privileges you can use SET ROLE TO … interactively
  - SQL compliant which is good
  - feels like sudo
  - Unlikely to be usable if you have applications with pooled connections to a backend database they'll need the approach mentioned above

Roles

Accounts with multiple roles

- If you want all roles to be active at the same time
  - ALTER USER user'@'%' DEFAULT ROLE ALL
  - SET DEFAULT ROLE TO ALL FOR 'user'@'%'
  - Typical for application users where you want to combine subsets of permissions

Roles

Accounts with multiple roles

- Output of SHOW GRANTS FOR `user`@`host_part` show a role as **role@%** rather than just **role**
    - Syntax matters and a role does not have a host part so showing it makes no sense.

To see the full grants for a user where a role is involved you need to do SHOW GRANTS FOR `user`@`host_part USING role_name. (note: no @%).

- This shows the derived grants for role_name but still keeps showing the GRANT `role_name` TO `user`@`host_part` . This does not seem consistent.
- **Missing:** no option for SHOW GRANTS FOR `user`@`host` USING ALL to provide a complete “role free” view of rights as if roles were not being used
- **Missing:** a way to see the final, derived (non role-based) grants

Roles

In spite of previous comments roles are good and most welcome

Replication

MySQL 8.0 (through GR) brought the new **WRITESET** method for tracking changes on the master and using this to replicate.

- Also backported to 5.7 which is really nice
- We have previously tested this and seen the behavior is much better than the traditional LOGICAL_CLOCK method available earlier
- Consider changing the configuration to use WRITESET if you suffer replication delays due to insufficient parallelism.

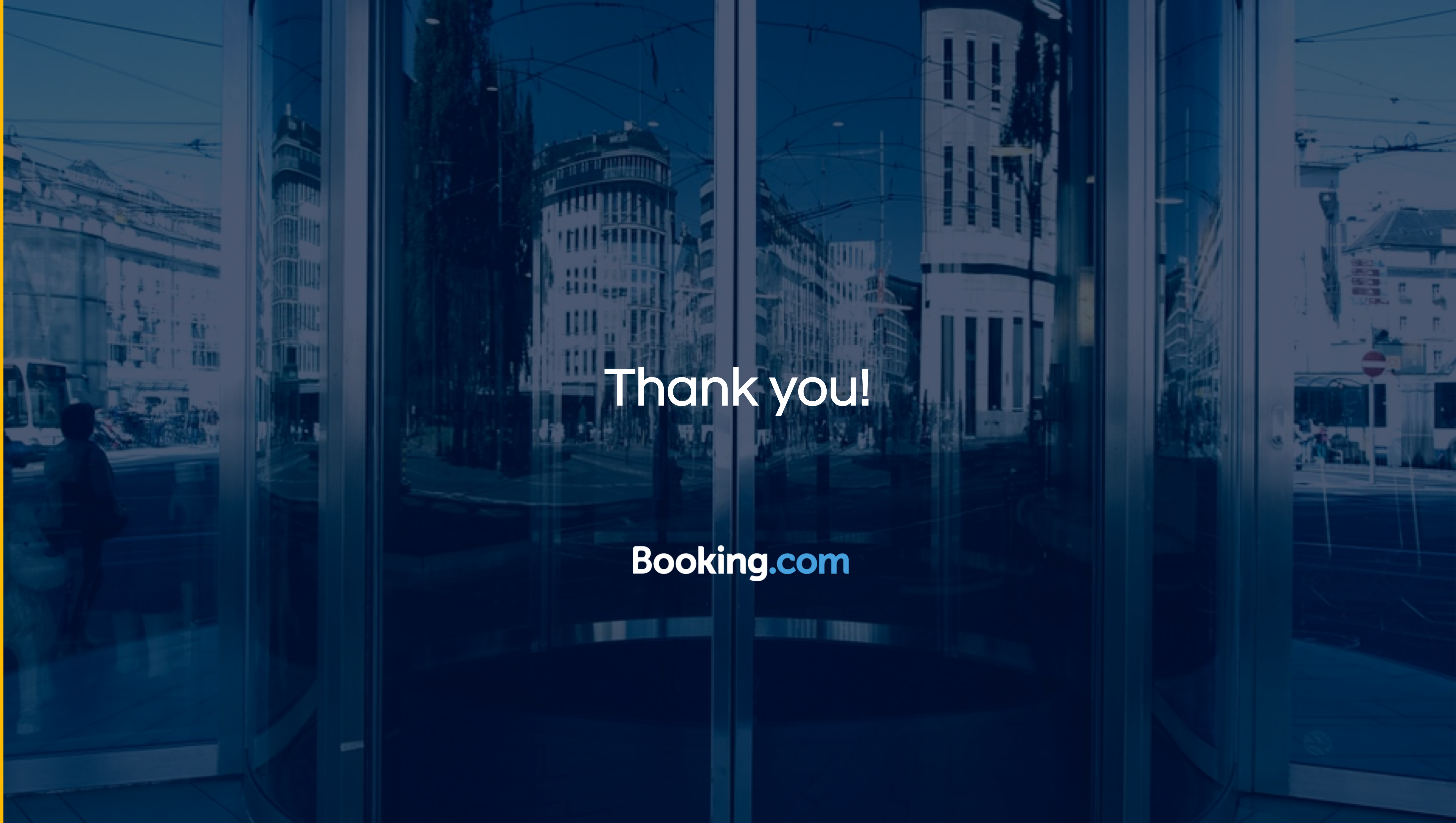
CTEs and GIS

These are different things and are most welcome

- In both cases there’s a matter of education of existing users that this functionality exists so that it can be evaluated and used
    - **Change in mindset required:** MySQL has traditionally been “simple” so when these new features come out even if they’re good they may get ignored

Conclusion

Despite all the previous ifs and buts and comments, MySQL 8.0 has shown to be extremely stable and provides a lot of new features that we really want!



Thank you!

Booking.com