

Database Access and ORM

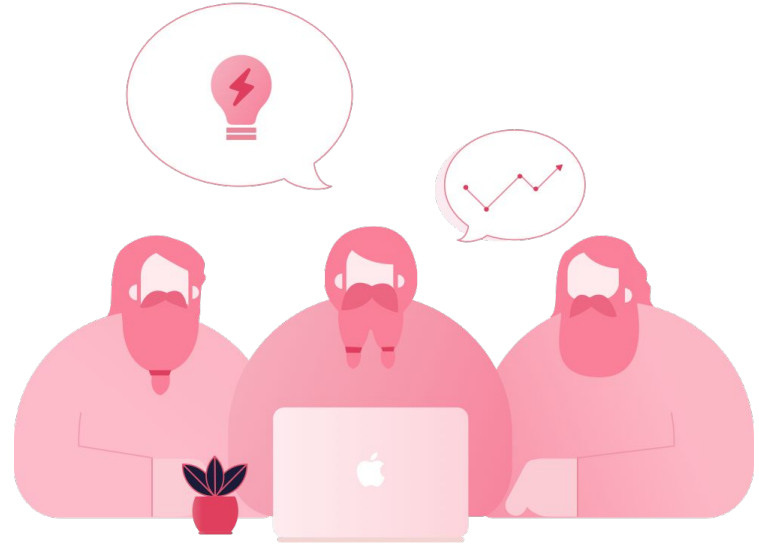


Hieu Phan

Engineer Lead

@hieupq

andy@dwarvesv.com



Agenda

Topic summary

1. SQL Databases and Executing Queries
2. Object-Relational Mapping (ORM)
3. GORM and XORM in Golang
4. Demo

SQL Databases and Executing Queries

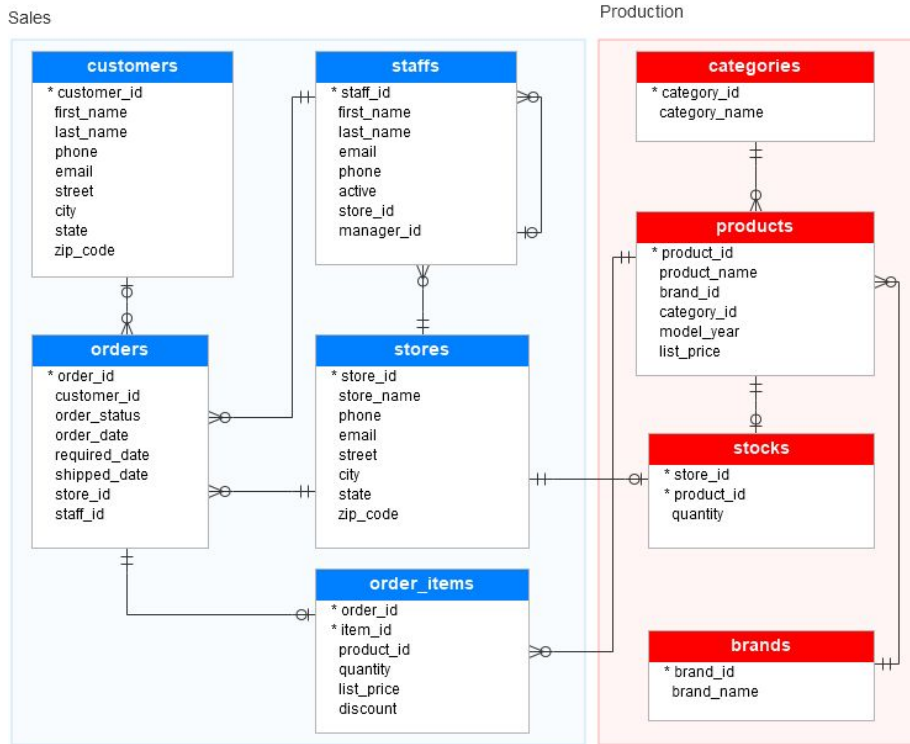
SQL Databases

- SQL stands for Structured Query Language
- SQL became a standard language in 1986 by the American National Standards Institute (ANSI) and later adopted by the International Organization for Standardization (ISO)



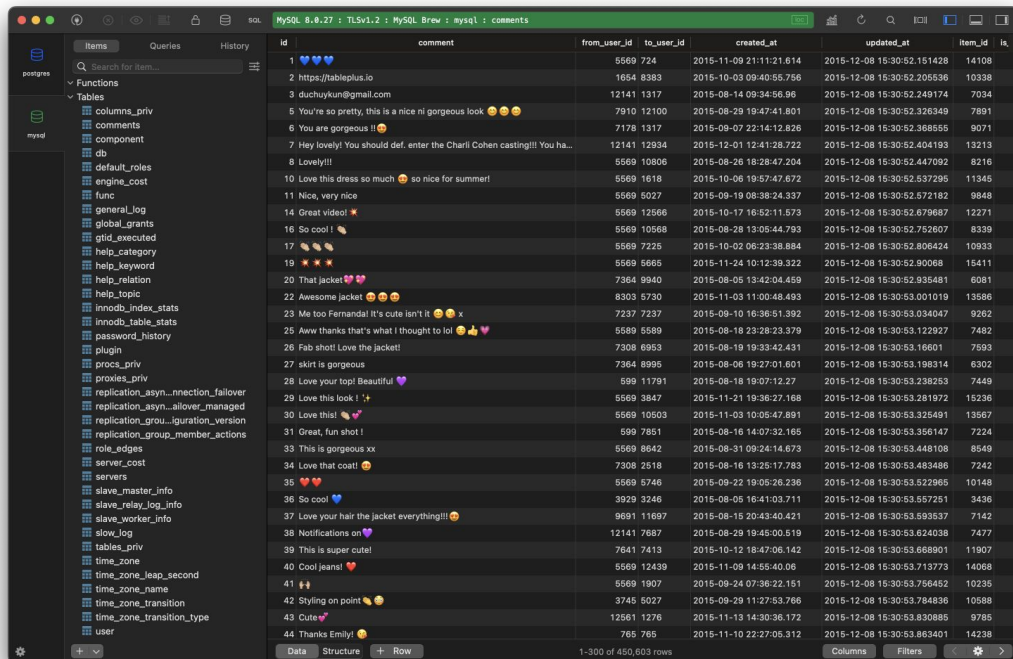
SQL Databases

- SQL Databases are often Relational Database Management Systems
- Data is mapped by 1-to-1, 1-to-many, and many-to-many relationships
- Relationships can be used to help keep data integrity and for joining data together



Executing SQL

- SQL is a standard language for managing and querying data in RDBMS
- Use tools like Beekeeper Studio or SQL or TablePlus to execute queries



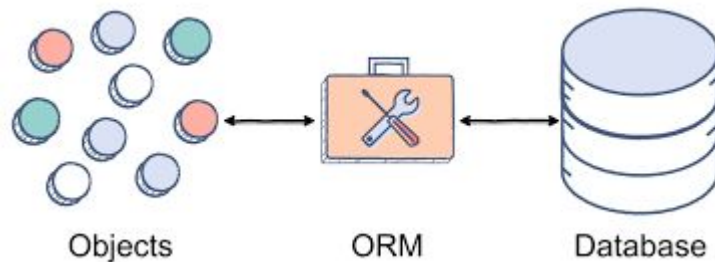
The screenshot shows the Beekeeper Studio interface with a MySQL database named 'mysql' and a table named 'comments'. The table has columns: id, comment, from_user_id, to_user_id, created_at, and updated_at. The table contains 45 rows of data, showing various user comments and their timestamps.

id	comment	from_user_id	to_user_id	created_at	updated_at
1	❤️❤️❤️	5569	724	2015-11-09 21:11:21.614	2015-12-08 15:30:52.151428
2	https://tableplus.io	1654	8383	2015-10-03 09:40:55.756	2015-12-08 15:30:52.205536
3	duchuykun@gmail.com	12141	1317	2015-08-14 09:34:56.96	2015-12-08 15:30:52.249174
5	You're so pretty, this is a nice ni gorgeous look 🥰🥰	7910	12100	2015-08-29 19:47:41.801	2015-12-08 15:30:52.326349
6	You are gorgeous it 🥰	7178	1317	2015-09-07 22:14:12.826	2015-12-08 15:30:52.368555
7	Hey lovely! You should def. enter the Charl Cohen casting!!! You ha...	12141	12934	2015-12-01 12:41:28.722	2015-12-08 15:30:52.404193
8	Lovely!!!	5569	10806	2015-08-26 18:28:47.204	2015-12-08 15:30:52.447092
10	Love this dress so much 🥰 so nice for summer!	5569	1618	2015-10-06 19:57:47.672	2015-12-08 15:30:52.537295
11	Nice, very nice	5569	6027	2015-09-19 08:38:24.337	2015-12-08 15:30:52.572182
14	Great video! 🌟	5569	12566	2015-10-17 16:52:11.573	2015-12-08 15:30:52.679687
16	So cool !	5569	10568	2015-08-28 13:05:44.793	2015-12-08 15:30:52.752607
17	🥰🥰🥰	5569	7225	2015-10-02 08:23:38.884	2015-12-08 15:30:52.808424
19	🌟🌟🌟	5569	5665	2015-11-24 10:12:39.322	2015-12-08 15:30:52.80068
20	That jacket! 🥰🥰	7364	8940	2015-08-05 13:42:04.459	2015-12-08 15:30:52.895481
22	Awesome jacket 🥰🥰🥰	8303	5730	2015-11-03 11:00:48.493	2015-12-08 15:30:53.001019
23	Me too Fernando! It's cute isn't it 🥰🥰 x	7237	7237	2015-09-10 16:36:51.392	2015-12-08 15:30:53.034047
25	Aww thanks that's what I thought to lol 🥰🥰🥰	5569	5589	2015-08-18 23:28:23.379	2015-12-08 15:30:53.122927
26	Fab shot! Love the jacket!	7308	6953	2015-08-19 19:33:42.431	2015-12-08 15:30:53.16601
27	skirt is gorgeous	7364	8995	2015-08-06 19:27:01.601	2015-12-08 15:30:53.198314
28	Love your top! Beautiful 🥰	599	11791	2015-08-18 19:07:12.27	2015-12-08 15:30:53.238253
29	Love this look ! 🌟	5569	3847	2015-11-21 19:36:27.168	2015-12-08 15:30:53.281972
30	Love this! 🥰🥰	5569	10503	2015-11-03 10:05:47.891	2015-12-08 15:30:53.325491
31	Great, fun shot !	599	7851	2015-08-16 14:07:32.165	2015-12-08 15:30:53.356147
33	This is gorgeous xx	5569	8642	2015-08-31 09:24:14.673	2015-12-08 15:30:53.448108
34	Love that coat! 🥰	7308	2518	2015-08-16 13:25:17.783	2015-12-08 15:30:53.483486
35	🥰🥰	5569	5746	2015-09-22 19:05:26.236	2015-12-08 15:30:53.522965
36	So cool 🥰	3929	3246	2015-08-06 16:41:03.711	2015-12-08 15:30:53.557251
37	Love your hair the jacket everything!!! 🥰	9691	11697	2015-08-19 20:43:40.421	2015-12-08 15:30:53.593537
38	Notifications on 🥰	12141	7687	2015-08-29 19:45:00.519	2015-12-08 15:30:53.624038
39	This is super cute!	7641	7413	2015-10-12 18:47:06.142	2015-12-08 15:30:53.668901
40	Cool jeans! 🥰	5569	12439	2015-11-09 14:55:40.06	2015-12-08 15:30:53.713773
41	🌟	5569	1907	2015-09-24 07:36:22.151	2015-12-08 15:30:53.756452
42	Styling on point 🥰	3745	5027	2015-09-29 11:27:53.766	2015-12-08 15:30:53.784836
43	Cute! 🥰	12561	1276	2015-11-13 14:30:36.172	2015-12-08 15:30:53.830885
44	Thanks Emily! 🥰	765	765	2015-11-10 22:27:05.312	2015-12-08 15:30:53.863401

Object-Relational Mapping (ORM)

ORMs

- Object-Relational Mapping (ORM) is a technique used in software development to create a bridge between object-oriented programs and relational databases
- ORM tools automate the process of mapping objects to database tables, handling database operations, and managing the persistence of data



GORM and XORM in Golang

GORM and XORM

- GORM follows a code-first approach, where you define Go structs as models and GORM automatically generates the corresponding database tables[3].
- It provides functionalities for schema auto-migration, logging, contexts, prepared statements, associations, constraints, advanced database operations like sharding, and more.
- XORM follows a join function design approach, which allows you to perform database operations with less code.
- It provides features like cache support, transactions, optimistic locking, multiple database support, and reverse engineering tools.



GORM

- We first open a connection to a sqlite database file. Then use AutoMigrate to create the schema for the Product model.
- To create a new product, we can use db.Create() and pass a Product struct.
- To read a product, we use db.First() which will find the product by primary key ID.
- We can then print the product details.
- This covers the basic CRUD operations for database interactions using GORM in Go.

```
1 package main
2
3 import (
4     "gorm.io/driver/sqlite"
5     "gorm.io/gorm"
6 )
7
8 type Product struct {
9     gorm.Model
10    Code string
11    Price uint
12 }
13
14 func main() {
15     db, err := gorm.Open(sqlite.Open("test.db"), &gorm.Config{})
16     if err != nil {
17         panic("Failed to connect to database")
18     }
19
20     // Migrate schema
21     db.AutoMigrate(&Product{})
22
23     // Create product
24     db.Create(&Product{Code: "D42", Price: 100})
25
26     // Read product
27     var product Product
28     db.First(&product, 1)
29
30     // Print product
31     println(product.Code) // D42
32 }
```

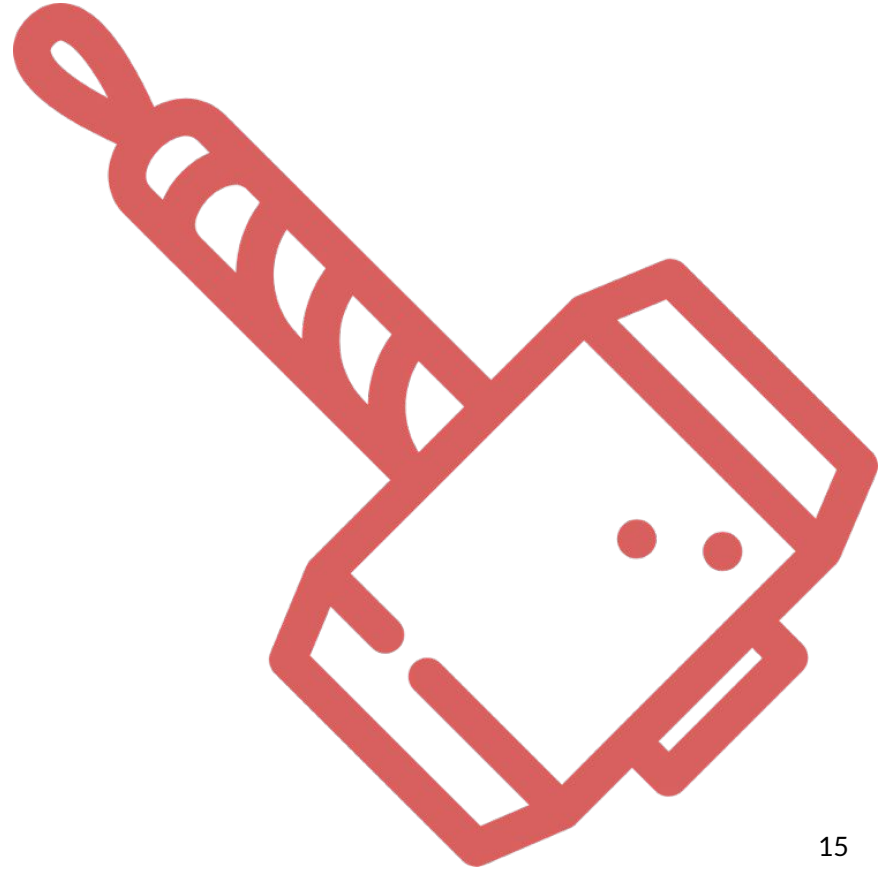
XORM

- We use Sync2() to migrate the schema for the Product model.
- To insert a new product, we can use engine.Insert() and pass the Product struct.
- To query, we use engine.ID(id).Get() to select by primary key.
- The main differences from GORM are the function names like Insert vs Create, and using Get() vs First() for querying. But overall very similar CRUD principles.

```
1 package main
2
3 import (
4     "fmt"
5
6     _ "github.com/mattn/go-sqlite3"
7     "xorm.io/xorm"
8 )
9
10 type Product struct {
11     Id      int64  `xorm:"pk autoincr"`
12     Code    string `xorm:"varchar(255)"`
13     Price   int64
14 }
15
16 func main() {
17     engine, err := xorm.NewEngine("sqlite3", "test.db")
18     if err != nil {
19         panic(err)
20     }
21
22     // Migrate schema
23     engine.Sync2(new(Product))
24
25     // Insert product
26     product := Product{Code: "D42", Price: 100}
27     _, err = engine.Insert(&product)
28
29     // Query product
30     var p Product
31     has, err := engine.ID(1).Get(&p)
32
33     fmt.Printf("Product: %v\n", p) // Product: {1 D42 100}
34 }
```

Demo

Demo - Zer0 to Hero



Reference

Resources & Reference links

- <https://gorm.io/>
- <https://xorm.io/>

Thank You



Q&A

