

فاز دوم پروژه (طراحی منطقی)
طراحی پایگاه داده‌ها

آرمان بابایی - ۹۸۱۰۰۳۴۹
حمیدرضا دهباشی - ۹۸۱۰۵۷۶۲
نوید اسلامی - ۹۸۱۰۰۳۲۳

۳ دی ۱۴۰۰

طراحی منطقی صورت گرفته

در این فاز، تصمیم بر این شد که با استفاده از ابزار MySQL Workbench طراحی منطقی را انجام دهیم. شکل کلی جداولی که طراحی شدند را می‌توانید در فایل Logical_Design.pdf ببینید. اما در این شکل، نمی‌شد به راحتی دیدهای ساخته شده را نمایش داد. به همین دلیل، تعریف دیدهای مربوطه را در ادامه آورده‌ایم. نام این دیدها را اما در فایل ذکر شده می‌توانید مشاهده کنید. همچنین، فایل مدل طراحی صورت گرفته را نیز در Logical_Model.mwb می‌توانید مشاهده کنید.

AccountBalance

```
1 • CREATE VIEW `AccountBalance` AS SELECT AccountID, Value_ - Paid AS Balance
2 FROM Account_ JOIN AccountTransactionSummary ON Account_.AccountID = AccountTransactionSummary.AccountID
3 JOIN AccountOrderSummary ON Account_.AccountID = AccountOrderSummary.AccountID
```

AccountInfo

```
1 • CREATE VIEW `AccountInfo` AS SELECT AccountId, FirstName, LastName, DateCreate, Email FROM Account
```

AccountOrderSummary

```
1 • CREATE VIEW `AccountOrderSummary` AS SELECT Account_.AccountID, SUM(OrderSummary.Price) AS Paid
2 FROM Account_ JOIN OrderSummary ON Account_.AccountID = OrderSummary.AccountID
3 GROUP BY OrderSummary.AccountID
```

AccountTransactionSummary

```
1 • CREATE VIEW `AccountTransactionSummary` AS SELECT Account_.AccountID, SUM(Transaction_.Value_) AS Value_
2 FROM Account_ JOIN Transaction_ ON Transaction_.AccountID = Account_.AccountID
3 GROUP BY Transaction_.AccountID
```

CartSummary

```
1 • CREATE VIEW `CartSummary` AS SELECT Cart.CostumerID, Cart.DateLastEdit, SUM(Product.Price * CartContainsProduct.Quantity)
2 AS Price
3 FROM
4 Cart JOIN CartContainsProduct ON Cart.CostumerID = CartContainsProduct.CartID
5 JOIN Product ON Product.ProductID = CartContainsProduct.ProductID
6 GROUP BY Cart.CartID
```

CustomerDiscount

```
1 • CREATE VIEW `CustomerDiscount` AS SELECT CustomerID, DiscountID, Code_, DateCreate, DateEnd, MaxDisAmount, MaxDisPercent,
2 MaxUsage, CurrentUsageCount, MaxUsage - CurrentUsageCount AS RemainingUsage
3 FROM DiscountCode JOIN Uses ON DiscountCode.DiscountID = Uses.DiscountCodeID
```

OrderSummary

```
1 • CREATE VIEW `OrderSummary` AS SELECT OrderSummaryByProduct, (SELECT MAX(Price_)
2 FROM (VALUES
3 ROW(SUM(Product.Price * OrderSummaryByProduct.Quantity) * DiscountCode.MaxDisPercent),
4 ROW(SUM(Product.Price * OrderSummaryByProduct.Quantity) - DiscountCode.MaxDisAmount)) AS AllPrices(Price_)
5 AS Price
6
7 FROM OrderSummaryByProduct JOIN Product ON Product.ProductID = OrderSummaryByProduct.ProductID
8 RIGHT JOIN DiscountCode ON DiscountCode.DiscountID = OrderSummaryByProduct.DiscountCodeID
9 GROUP BY OrderSummaryByProduct.OrderID
```

OrderSummaryByProduct

```
1 • CREATE VIEW `OrderSummaryByProduct` AS SELECT Order_.OrderID, SubOrderHasProduct.ProductID, Order_.DiscountCodeID,
2 SUM(SubOrderHasProduct.Quantity) AS Quantity
3 FROM Order_ JOIN SubOrder ON Order_.OrderID = SubOrder.OrderID
4 JOIN SubOrderHasProduct ON SubOrderHasProduct.SubOrderID = SubOrder.SubOrderID
5 GROUP BY SubOrderHasProduct.ProductID
```

OrderTransfers

```
1 CREATE VIEW `OrderTransfers` AS SELECT Transfer.TransferID, CustomerID, Fee, Date, DeliveryID, FromStorageID, T0AddressID,
2 OrderID
3 FROM Transfer JOIN Order_ ON Transfer.OrderID = Order_.OrderID
```

ProductRate

```
1 • CREATE VIEW `ProductRate` AS SELECT ProductID, AVG(Rate)
2 FROM Product JOIN Review ON Product.ProductID = Review.ProductID
3 GROUP BY ProductID
```

SubOrderTransfers

```
1 • CREATE VIEW `SubOrderTransfers` AS SELECT Transfer.TransferID, StorageID, Fee, Date, DeliveryID, FromStorageID, T0AddressID,
2 SubOrder.SubOrderID
3 FROM Transfer JOIN SubOrder ON Transfer.SubOrderID = SubOrder.SubOrderID
```

تغییرات صورت گرفته در EER

در این فاز، مشکلاتی در نمودار EER مشخص شد که آن‌ها را برطرف نمودیم. نمودار EER جدید را در فایل‌های EER_Entities_and_Attributes.png و EER_Relations.png می‌توانید مشاهده کنید.

انتخاب سیستم مدیریت پایگاه داده

با توجه به بررسی‌های انجام‌شده، تصمیم بر آن شد که از MySQL به عنوان DBMS پروژه استفاده شود. این سیستم پایگاه داده تحت توسعه‌ی شرکت Oracle است. MySQL واسط‌های متنوعی برای زبان‌های Perl، Python، PHP، C/C++ دارد و می‌تواند تحت پلتفرم‌های ویندوز، لینوکس، مک‌اواس

و ... استفاده شود. این سیستم پایگاه داده یک نسخه‌ی رایگان برای استفاده‌ی شخصی به کاربران خود ارائه می‌دهد که حتی برای کسب‌وکارهای کوچک کفایت می‌کند.

این سیستم پایگاه داده ابزارهای مناسبی برای ایجاد تغییرات با دستورات ساده ارائه می‌کند. به همین دلیل یادگیری کار با این سیستم پایگاه داده ساده است. علاوه بر این ابزارهای عرضه‌کننده‌ی ابری از این سیستم پایگاه داده پشتیبانی می‌کنند. علاوه بر این، MySQL موتورهای متنوعی برای ذخیره‌سازی جداولش از جمله MyISAM، InnoDB، Merge، ARCHIVE، CSV، FEDERATED ارائه می‌دهد. (هرچند استفاده از برخی از این موتورها منجر به محدودیت‌هایی می‌شود؛ مثلاً اگر از موتور FEDERATED استفاده شود، امکان تعریف کلید خارجی در جداول از بین می‌رود)

از طرفی، این پایگاه داده برای گسترش در هنگام بزرگ شدن کسب‌وکار، برنامه‌نویس را با سختی‌های زیادی روبرو می‌کند به طوری که اغلب کاربران این سیستم پایگاه داده با بزرگ شدن کاربرد خود، به سیستم دیگری (معمولاً پایگاه‌های داده‌ی غیررابطه‌ای) مهاجرت می‌کنند.

همچنین علیرغم این که این سیستم پایگاه داده متن‌باز است، تحت توسعه‌ی شرکت Oracle قرار دارد و جامعه‌ی برنامه‌نویسانی که از آن استفاده می‌کنند نمی‌توانند تغییرات مورد نیاز خود را به طور مستقیم اعمال کنند و در نتیجه روال طولانی برای ایجاد تغییرات دلخواه کاربران در این سیستم پایگاه وجود دارد. برای رفع این ایراد MariaDB معرفی شده که شاخه‌ای از MySQL است که تحت لیسانس GNU است. علاوه بر این موارد، یکی از ایرادهایی که می‌توان بر MySQL گرفت، این است که این سیستم پایگاه داده، به طور کامل از استانداردهای SQL پشتیبانی نمی‌کند و در برخی موارد با آن تفاوت‌هایی دارد. به طور مثال MySQL از دستور `SELECT ... INTO TABLE` پشتیبانی نمی‌کند.

با توجه به آنچه بیان شد، MySQL با توجه به سادگی و جامعه‌ی کاربران بزرگ، یک انتخاب مناسب برای انجام پروژه‌ی دانشگاهی است.