

به نام خدا

طراحی منطقی پایگاه داده

کیوان رضائی و سیدمحمد سیدجوادی

فاز ۱ و ۲:

DBMS مورد استفاده ما برای فازهای بعد همان **postgresql** است که در کلاس نیز با آن کار می کردیم.

Tables:

- Customer (NC, First_Name, Last_Name, Money)
- Comment (Customer_NC, Comment_No, Text)
- Discount (Customer_NC, Discount_No, Percent, Expiration_Time, Order_No, Customer_Order_NC)
- Airplane (Code, Capacity, Model, City)
- Employee (Code, Name , Job_Type, Employment_Year, Salary, Total_Salary)
- Seat (Airplane_Code, Seat_No)
- Order (Customer_NC, Order_No, Payment_Status, Travel_Code, Score, Seat_No, Airplane_Code)
- Travel (Code, Time, Starting_City, Target_City, Ticket_Price, Airplane_Code, Captain_Code)
- Fligh_Crew(Employee_Code, Travel_Code)

دیدهای خارجی:

جداول View جدید با رنگ آبی مشخص شده اند. برخی جداول اصلی از دید کاربران مختلف مخفی شده است.

دیدهای از سمت برنامه ساز نیازمندی های کاربر:

- Customer (NC, First_Name, Last_Name, Money)
- Comment (Customer_NC, Comment_No, Text)
- Discount (Customer_NC, Discount_No, Percent, Expiration_Time, Order_No, Customer_Order_NC)

- Order (Customer_NC, Order_No, Payment_Status, Travel_Code, Score, Seat_No, Airplane_Code)
- Travel (Code, Time, Starting_City, Target_City, Ticket_Price, Airplane_Code, Captain_Code)
- *Airplane_Score* (Airplane_Code, Score)
- *Travel_Empty_Seats* (Travel_Code, Seat_No, Airplane_Code)

دیدهای از سمت برنامه‌ساز نیازمندی‌های مدیر:

- همه جدول‌ها
- *Airplane_Score* (Airplane_Code, Score)
- *Crew_Score* (Employee_Code, Score)
- *Captain_Score* (Employee_Code, Score)

در طراحی، View هایی که روی جدول هستند را به User های پایگاه داده GRANT دادیم.

فاز ۳

دامنه‌ها و محدودیت‌های صفات:

ضمناً صفاتی که کلید هستند طبیعتاً not null و unique هستند که برای کوتاه‌تر شدن نیاورده‌ایم. همچنین foreign key ها نیز با خط‌چین مشخص شده‌اند.

- Customer (NC, First_Name, Last_Name, Money)
NC, password, First_Name, Last_Name **varchar**
Money **real not null** (Money >= 0)
- Comment (Customer_NC, Comment_No, Text)
Customer_NC, Comment_NO, text **varchar**
- Discount (Customer_NC, Discount_No, Percent, Expiration_Time, Order_No, Customer_Order_No)

Percent **real not null** ($0 \leq \text{Percent} \leq 100$)

Expiration_Time **timestamp not null**

Customer_NC, Discount_No, Order_No **varchar**

صفت customer_order_NC چون تابعی از customer_NC (زیرمجموعه‌ای از کلید اصلی) بود جهت نرمال‌تر سازی به فرم 2NF حذف شد.

- Airplane (Code, Capacity, Model, City)
Code, Model, City **varchar not null**
Capacity **integer not null** (capacity ≥ 0)
- Employee (Code, Name, Job_Type, Employment_Year, Salary, Total_Salary)
Code, Name, Employment_Year **varchar**
Job_Type **varchar not null**
Salary, Total_Salary **real not null** (Salary, Total_Salary ≥ 0)
- Seat (Airplane_Code, Seat_No)
Airplane_Code, Seat_No **varchar**
- Order (Customer_NC, Order_No, Payment_Status, Travel_Code, Score, Seat_No, Airplane_Code)
Customer_NC, Order_No, Travel_Code, Seat_No **varchar not null**
Payment_Status **varchar not null** (in ('Paid', 'NotPaid'))
Score **double** ($0 \leq \text{score} \leq 100$)

صفت Airplane_Code را چون تابعی از Travel_Code بود و یک FD با واسطه ایجاد می‌کرد، به دلیل نرمال‌تر سازی جدول به فرم 3NF حذف کردیم.

- Travel (Code, Time, Starting_City, Target_City, Ticket_Price, Airplane_Code, Captain_Code)
Code, airplaneCode, startCity, targetCity, captainCode **varchar not null**
Time **timestamp not null**
ticketPrice **double not null** (ticketPrice ≥ 0)

- `Fligh_Crew(Employee_Code, Travel_Code)`
`Employee_Code, Trvavel_Code varchar`

* البته در فایل `TableDefiner.py` در فولدر `Definer` این محدودیت‌ها در تعریف جداول و دامنه‌ها به طور کامل مشاهده می‌شود.

رہانا

۶ رہانا (Trigger) تعریف شده است که جزئیات کامل پیاده‌سازی آن‌ها در فایل `Initializer.py` و در تابع `triggers` وجود دارد. در ادامه توضیح هر یک از آن‌ها آمده است.

- رہانایی که چک می‌کند مدیر، هنگام افزودن یک پرواز، پروازی با زمان قبل از زمان فعلی درج نکند. یعنی پروازی برای زمان آینده اضافه کند. (نام رہانا `insert_travel_time`)
- رہانایی که چک می‌کند مدیر، شماره کارمندی که به عنوان خلبان پرواز تعیین می‌کند واقعا یک خلبان باشد و از میان کارمندان دیگر (که می‌توانند عناوین مختلفی داشته باشند) نباشد. (نام رہانا `insert_travel_captain`)
- رہانایی که افزودن خدمه پرواز از سمت مدیر، پیش از آغاز پرواز انجام گیرد. یعنی مدیر نتواند پس از اتمام یک پرواز کارمندی به آن اضافه کند. (نام رہانا `insert_flight_crew`)
- رہانایی که چک می‌کند مدیر هنگام دادن تخفیف به کاربران، تخفیفی با `expiration date` قدیمی‌تر از زمان حال، به کسی ندهد (نام رہانا `insert_discount`)
- رہانایی که چک می‌کند مجموع حقوق کارمندان همواره زیادت‌ر شود و هیچگاه کم نشود (نام رہانا `increasing_total_salary`)
- رہانایی که چک می‌کند هواپیمایی که مدیر برای یک سفر در نظر می‌گیرد، در شهر مبدا آن سفر مستقر باشد. (نام رہانا `insert_travel_airplane_city`)

نرمال‌تر سازی

در این طراحی ما هیچ صفت چندمقداری نداریم، بنابراین همه جداول در فرم نرمال 1NF هستند. از طرفی، در همه جداول FD میان کلید اصلی و صفات ناکلید از نوع کامل است و هیچ زیرمجموعه‌ای از صفات کلید اصلی یک FD با بقیه صفات نمی‌سازد.

(تغییراتی در جداول ایجاد کردیم که به این خاصیت برسیم. مثلاً حذف Customer_Order_NC از جدول Discount که تابعی از فقط CustomerNC بود)
از طرفی هیچ FD با واسطه‌ای میان صفات ناکلید و کلید اصلی وجود ندارد، لذا رابطه 3NF هم هست.
(باز هم تغییراتی در جداول ایجاد کردیم که به این خاصیت برسیم، مثلاً حذف Airplane_Code از جدول Order که تابعی از Travel_Code است که سبب می‌شد FD با واسطه شکل بگیرد)

کار با سامانه

برای کار با سامانه، نخست فایل Initializer.py را اجرا کنید. برای این موضوع لازم است در فایل Login.py، متغیر my_database را نام دیتابسی که می‌خواهید استفاده کنید، متغیر manager_username را نام اکانت postgres و manager_password را پسورد اکانت postgres تان قرار دهید.

برای اجرای اپلیکیشن به عنوان مدیر، فایل Manager.py را اجرا کنید که در ادامه به کمک ترمینال با پایگاه داده در ارتباط خواهید بود.
برای اجرای اپلیکیشن به عنوان یک کاربر، فایل User.py را اجرا کنید و مجدداً به کمک ترمینال با پایگاه داده در ارتباط خواهید بود.

کوئری‌ها در دو فایل ManagerQueries.py و UserQueries.py موجودند. همچنین بخشی از کوئری‌های کاربر User در فایل UserTerminal.py است.

تمرکز افراد گروه:

در فازهای اول و دوم با هم جلسه می‌گذاشتیم و کل کار را با هم جلو بردیم.
در فاز سوم، یک بخش تعریف جداول و دیدها بود که کار را دو دسته کردیم.
دسته ۱ (رضائی):

customer, comment, employee, travel, flight crew
دسته ۲ (سیدجوادی):

discount, airplane, seat, order
دید مدیر

برای قسمت ترمینال‌ها و کوئری‌ها در کل سیدجوادی بخش user را زد و رضائی بخش manager. ولی برای تست هر دو تمام قسمت‌ها را تست می‌کردیم، برای بخش initialize هم هر دو هر کدام قسمتی را زدیم.
در کل زمان زیادی در skype با هم کار را پیش بردیم.