

Bazy danych 1

Leszek Błazewski, 241264

Karol Noga, 241259

Semestr letni 2018/2019

1 Normalizacja bazy danych a redundancja

Proces normalizacji bazy danych polega na doprowadzaniu bazy do postaci w której w bazie nie występuje redundancja. Redundancja jest zjawiskiem na ogół niepożądanym, lecz w specyficznych przypadkach jest konieczna w celu zapewnienia jakości danych usług (np. redundancja serwerów). Rozważając redundancję ze względu na dane występujące w bazie zauważyć możemy, że zjawisko to nie przynosi żadnych korzyści a jedynie przechowuje dane, które składowane są już w bazie tym samym zajmując miejsce, które wykorzystane mogłoby być na nowe informacje. Do redundancji dochodzić może gdy relacje pomiędzy obiektami nie zostały dokładnie przemyślane.

2 Proces normalizacji

Podczas pierwszej implementacji, nie zwróciliśmy uwagi na relacje zachodzące pomiędzy encją pracownika pływalni a jego czasem oraz dniami pracy. Pierwszym pomysłem było przechowywanie wszystkich danych dotyczących danego pracownika (włącznie z czasem pracy, dniami tygodnia gdy dana osoba pracuje oraz basenem do którego jest przydzielona) w tabeli *STAFF*. Takie rozwiązanie powodowało znaczne zwiększenie rozmiaru rekordów, co niekorzystnie wpłynęłoby na czas przeszukiwania danej tabeli. Problematiczne byłoby również przeszukiwanie tabeli w celu znalezienia pracowników pracujących w danych dniach, ponieważ każdy z rekordów odpowiadających danemu pracownikowi przechowywałby listę dni tygodnia przetrzymywaną w jednej kolumnie *WORKINGDAYS*, co kłóci się z pierwszą postacią normalną (1NF).

Rozwiązaniem problemu jest wprowadzenie osobnej tabli *SCHEDULES*, która abstrahuje dane dotyczące czasu pracy oraz dni tygodnia od reszty informacji dotyczących pracownika. Tabela *SCHEDULES* przechowuje klucz obcy do pracownika oraz do basenu, dzięki czemu znacznie redukujemy liczbę kolumn w tabeli i korzystamy z relacji klucz obcy -> główny. Rozwiązanie to też w prosty sposób pozwoliło uniknąć problemu przetrzymywania listy dni w jednej kolumnie, dzięki czemu pierwsza zasada postaci normalnej zostaje zachowana.

Poniżej zamieszczono diagram ER wygenerowany przy pomocy narzędzia Oracle SQL Developer Data Modeler.

RESERVATIONS		
F *	CLIENTID	NUMBER (*,0)
F *	POOLID	NUMBER (*,0)
*	RESERVATIONDATE	DATE
*	STARTTIME	NUMBER (4,2)
*	ENDTIME	NUMBER (4,2)
*	PRICE	NUMBER (5,2)
FK_CLIENTID (CLIENTID)		
FK_POOLID (POOLID)		

CLIENTS		
P *	ID	NUMBER (*,0)
*	NAME	NVARCHAR2 (20)
*	SURNAME	NVARCHAR2 (20)
*	PERSONALIDENTITYNUMBER	NUMBER (11)
*	PHONENUMBER	VARCHAR2 (15 CHAR)
*	SWIMMINGSKILL	NUMBER (2)
*	AGE	NUMBER (3)
CLIENTS_PK (ID)		
CLIENTS_PK (ID)		

POOLS		
P *	ID	NUMBER (*,0)
*	NUMBEROFPLACES	NUMBER (*,0)
*	REQUIRESKILL	NUMBER (2)
*	SPOTPRICE	NUMBER (5,2)
POOLS_PK (ID)		
POOLS_PK (ID)		

SCHEDULES		
F *	STAFFID	NUMBER (*,0)
F *	POOLID	NUMBER (*,0)
*	STARTTIME	NUMBER (4,2)
*	ENDTIME	NUMBER (4,2)
*	DAYOFWEEK	VARCHAR2 (9 BYTE)
FK_POOLID_REF (POOLID)		
FK_STAFFID (STAFFID)		

STAFF		
P *	ID	NUMBER (*,0)
*	NAME	NVARCHAR2 (10)
*	SURNAME	NVARCHAR2 (15)
*	SALARY	NUMBER (7,2)
*	JOB	VARCHAR2 (10 BYTE)
F	SUPERVISOR	NUMBER (*,0)
STAFF_PK (ID)		
FK_SUPERVISOR (SUPERVISOR)		
STAFF_PK (ID)		

