Lab 4: SQL

Jake Dame
13 Jun 2024
CS 6016: Database Systems & Applications
Dr. Nabil Makarem

Part 1: SQL Command Line

n/a

Part 2: Creating the Dealership

See CADE database.

`Cars [ __VIN__ (integer), CarMake (string), CarModel (string), CarYear (integer), CarColor (string) ]`

```sql
CREATE TABLE Cars
(
    VIN CHAR(17) NOT NULL,
    CarMake VARCHAR(255) NOT NULL,
    CarModel VARCHAR(255) NOT NULL,
    CarYear SMALLINT UNSIGNED NOT NULL,
    CarColor VARCHAR(255) NOT NULL,
    PRIMARY KEY (VIN)
);
```

`SalesAssociates [ __SSN__ (integer), AssociateName (string) ]`

```sql
CREATE TABLE SalesAssociates
(
    SSN CHAR(9) NOT NULL,
    AssociateName VARCHAR(255) NOT NULL,
    PRIMARY KEY (SSN)
);
```

`[ __VIN__(integer), __SSN__(integer) ]`

```sql
CREATE TABLE SalesAssignments
(
    VIN CHAR(17) NOT NULL,
    SSN CHAR(9) NOT NULL,
    PRIMARY KEY (VIN, SSN),
    FOREIGN KEY (VIN) REFERENCES Cars (VIN)
    on update cascade
    on delete cascade,
    FOREIGN KEY (SSN) REFERENCES SalesAssociates (SSN)
    on update cascade
    on delete cascade
);
```

```
insert into Cars values
('a1a1a1a1a1a1a1a1a', 'Toyota', 'Tacoma', 2008, 'Red'),
('b2b2b2b2b2b2b2b2b', 'Toyota', 'Tacoma', 1999, 'Green'),
('c3c3c3c3c3c3c3c3c', 'Tesla', 'Model 3', 2018, 'White'),
('d4d4d4d4d4d4d4d4d', 'Subaru', 'WRX', 2016, 'Blue'),
('e5e5e5e5e5e5e5e5e', 'Ford', 'F150', 2004, 'Red');
```

```
insert into SalesAssociates values
('111111111', 'Arnold'),
('222222222', 'Hannah'),
('333333333', 'Steve');
```

```
insert into SalesAssignments values
('a1a1a1a1a1a1a1a1a', '111111111'),
('a1a1a1a1a1a1a1a1a', '222222222'),
('b2b2b2b2b2b2b2b2b', '111111111'),
('e5e5e5e5e5e5e5e5e', '222222222'),
('c3c3c3c3c3c3c3c3c', '333333333');
```

Part 3: Simple Retrieval Queries

1. "Get the ISBNs of all books by <Author>"

```
select ISBN from Titles where author=<Author>;
```

2. "Get Serial numbers (descending order) of all books by <ISBN>"

```
select Serial from Inventory where ISBN=<ISBN> order by Serial desc;
```

3. "Get the Serial numbers and ISBNs of all books checked out by <Patron’s name>"

```
select Serial,ISBN from CheckedOut
natural join Inventory
natural join Patrons
where name=<Patron's name>;
```

4. "Get phone number(s) and Name of anyone with <ISBN> checked out"

```
select Phone,Name from CheckedOut
natural join Inventory
natural join Patrons
natural join Phones
where ISBN=<ISBN>;
```

Part 4: Intermediate Retrieval Queries

1. "Find the Authors of the library's oldest <N> books. Assume the lowest serial number is the oldest book."

```
select Author from Inventory
natural join Titles
order by Serial asc limit <N>;
```

2. "Find the name and phone number of the person who has checked out the most recent book."

```
select Name,Phone from Patrons
natural join Phones
natural join CheckedOut
where Serial=(select max(Serial) from CheckedOut);
```

3. "Find the phone number(s) and name of anyone who has checked out any book."

```
select distinct Phone,Name from CheckedOut
natural join Patrons
natural join Phones;
```

4. "Find the Authors and Titles of the books who have NOT been checked out by anyone. The query should not return duplicates."

```
select distinct Author,Title from Titles
natural join Inventory
where Serial not in (select Serial from CheckedOut);
```

Part 5: Chess Queries

1. "Find the names and IDs of any player with the 10 highest Elo ratings."

```
select Name,pID from Players order by Elo desc limit 10;
```

2. "Find the names and Elo ratings of any player who has ever played a game as black."

```
select distinct Name,Elo from Players
natural join Games
where pID=BlackPlayer;
```

3. "Find the names of any player who has ever won a game as white."

```
select distinct Name from Players
natural join Games
where pID=WhitePlayer and Result='W';
```

4. "Find the names of any player who played any games between 2014 and 2018 in Budapest HUN ."

```
select distinct Players.Name from Players
join Games on pID=WhitePlayer or pID=BlackPlayer
join Events on Games.eID = Events.eID --this doesn't work unless you
disambiguate
where Date>='2014-01-01' and Date<='2018-12-31' and Site='Budapest HUN';
```

5. "Find the Sites and dates of any event in which Garry Kasparov won a game."

```
select distinct Site,Date from Events
natural join Games
join Players on
Players.Name='Kasparov, Garry' and
(pID=BlackPlayer and Result = 'B' or pID=WhitePlayer and Result = 'W');
```

6. "Find the names of all opponents of Magnus Carlsen. An opponent is someone who he has played a game against. Hint: Both Magnus and his opponents could play as white or black."

```
select Name from
(
select P1.Name
from Games
join Players P1 on P1.pID=BlackPlayer
join Players P2 on P2.pID=WhitePlayer
where P2.Name='Carlsen, Magnus' and P1.Name!='Carlsen, Magnus'
union
select P2.Name
from Games
join Players P1 on P1.pID=BlackPlayer
join Players P2 on P2.pID=WhitePlayer
where P1.Name='Carlsen, Magnus' and P2.Name!='Carlsen, Magnus'
)
as TheOps;
```