



NOTE: This was a semester-long group project using Microsoft SQL Server to create and utilize a SQL database for an imaginary client, meant to emulate real-world projects. Some of the key skills learned were:

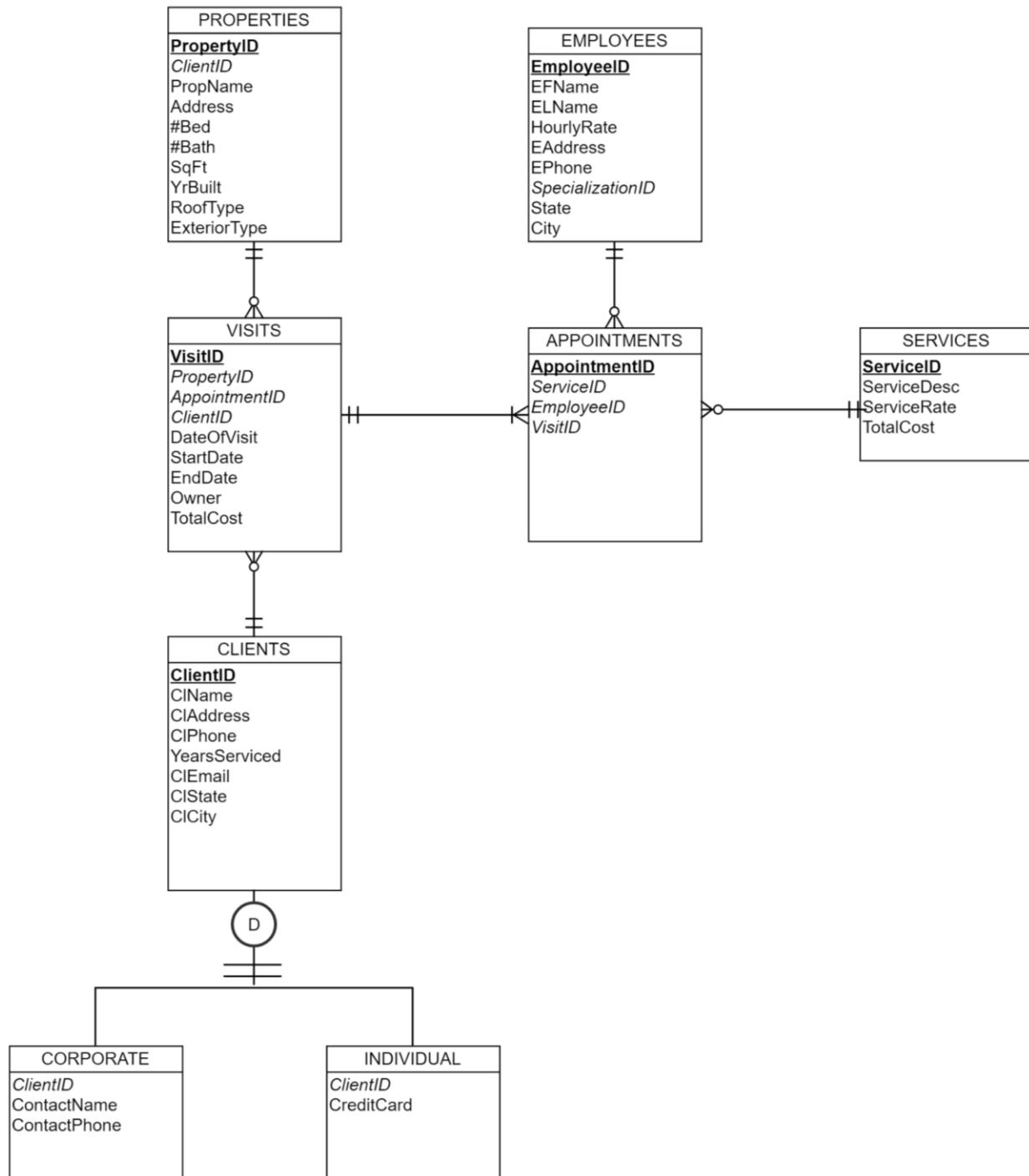
- SQL coding
- Creating a relational database
 - Integrity rules
- Basic and Advanced Querying
 - Query Optimization
- Data Visualization (using Tableau)
- Triggers, Procedures, Indexes, etc.
 - ERD's and Schema
 - And many more...

Executive Summary

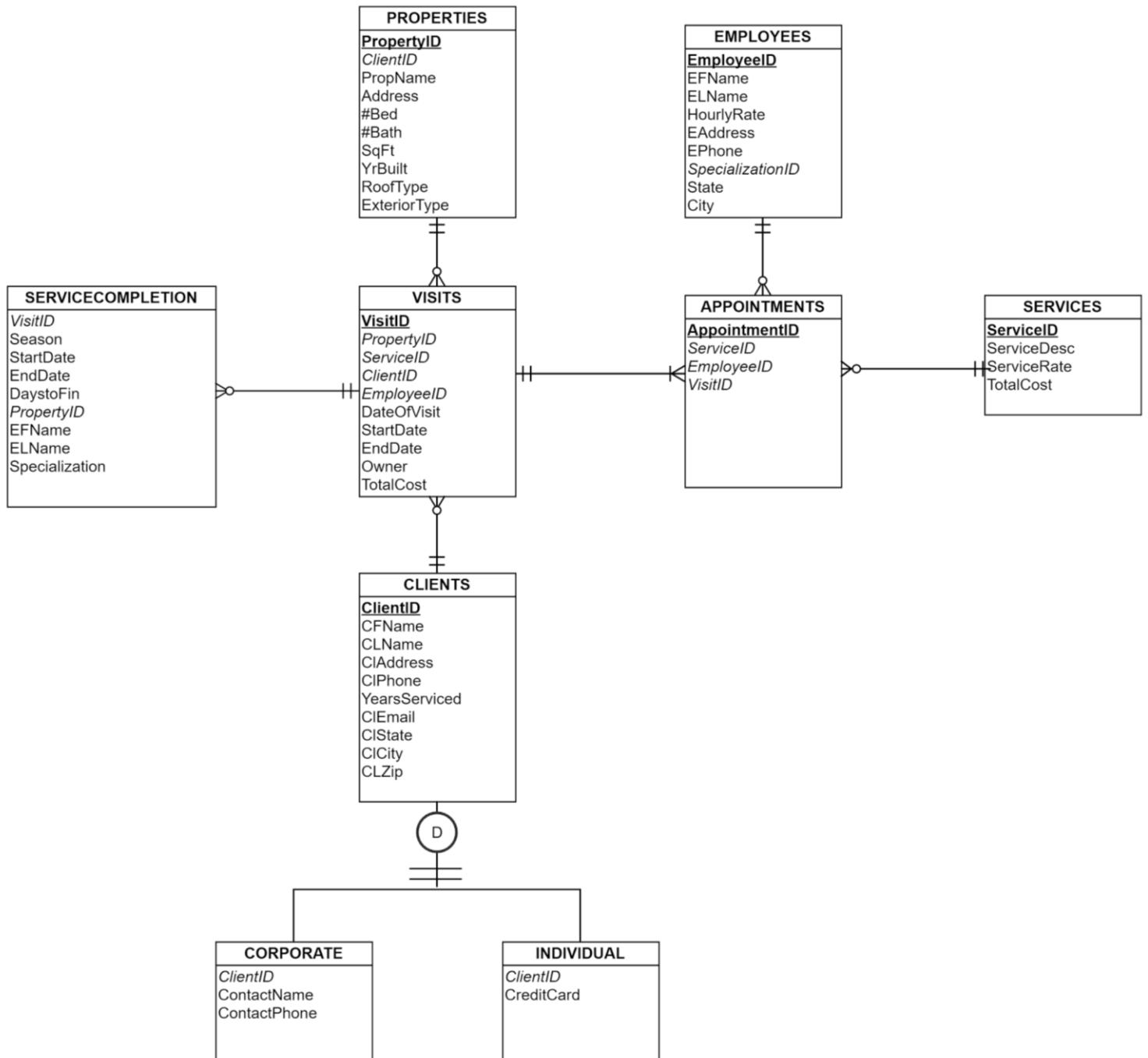
We began our project by creating an entity-relationship diagram and schema for the eight necessary tables that would be needed in our database. We were then able to use Mockaroo, a random data generator, to give us viable and reasonable data to insert into our tables. We then ran insert scripts in SQL Server to create our tables. After performing a variety of entity and referential queries to test our database, we were able to get things up and running. We were then able to run basic and advanced queries, transactions, triggers, procedures, and indexes to satisfy Eye On U's business needs. As we ran more and more detailed queries, we realized that our database would need to be modified in order to work properly, so we deemed the addition of a ServiceCompletion table to be necessary. One of the other primary issues that our group ran into was the loss of one of our team members. Our lost team member did not give us any heads up about his departure, so we had to stay agile in our approach and make up for the loss. Since he had some important documents, it set us back a little, but we persevered and were able to pick up the slack together. Our team relied on a number of Zoom meetings to stay in touch with each other, as the Coronavirus meant that we would no longer be able to meet up in-person to work on the milestones. Virtual communication became a very important part of our group, and we maintained strong communication through text group chats and weekly Zoom meetings. Each member of our team has different strengths and weaknesses and as we got further and further into the project, we were able to work more efficiently as a group by playing into our individual strengths. Each remaining member of our group played a vital role in the completion of the project and losing more members would have been extremely detrimental in the successful completion of the database. Going forward, our advice to other groups would be that it is extremely important to remain agile. A waterfall method for this project could cause severe problems, but if good communication and the ability and openness to change is maintained, then everything can and will work out!

Entity Relationship Diagram (ERD)

Original ERD:



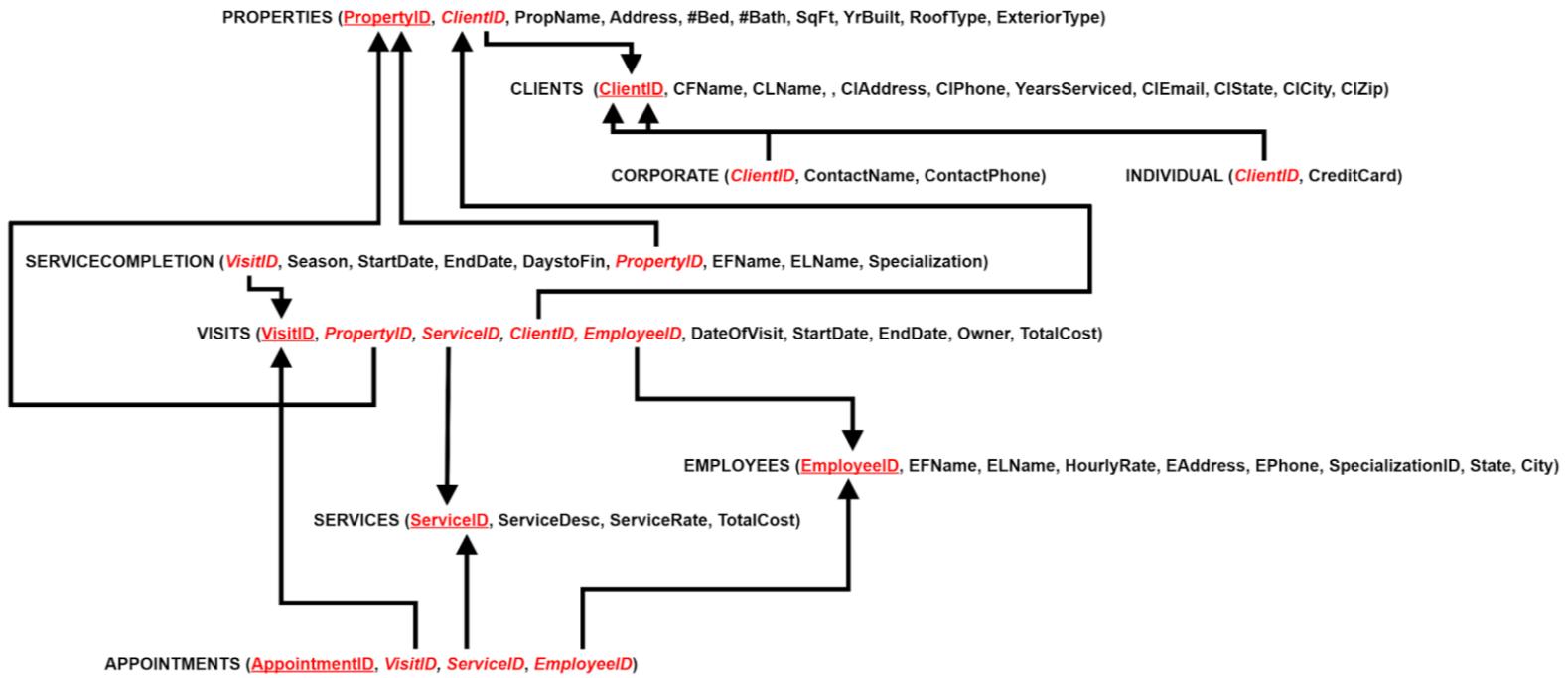
Finalized ERD:



What Changed?

We added a **ServiceCompletion** table, as well as ServiceID and EmployeeID to the **Visits** table and CFName, CLName, and CLZip to the **Clients** Table. The additions to the **Visits** table were to better track the Clients' names, as organizing by last name and first name separately is easier than having a combined value for both. Adding to the **Visits** table helped keep track of the appointments and linking the Employees and Services to each respective visit. The **ServiceCompletion** table was added to create the procedure in Milestone 5. This grabs the VisitID, StartDate, EndDate, and PropertyID from the Visits table and EFName, ELName, and Specialization from the Employees table using the corresponding EmployeeID from the Visits table.

Logical Relational Schema



Drafted from final ERD

Screenshots of Database Tables

EMPLOYEES Table

| | EmployeeID | EFName | ELName | HourlyRate | EAddress | EPhone | State | City | SpecializationID |
|----|------------|--------------|-------------|------------|--------------------------|------------|-------|----------------|------------------|
| 1 | 1000000000 | Test | Subject | 10.00 | 123 Main St | 3613762359 | TX | Corpus Christi | 1 |
| 2 | 111111 | Jake | Plum | 20.00 | 1234 Plumber Ave | 1234567890 | AZ | Phoenix | 10 |
| 3 | 1303503027 | Ulrikaumeiko | De Dantesie | 32.32 | 456 Killdeer Alley | 3235467518 | CA | Los Angeles | 1 |
| 4 | 2075868651 | Mallorie | Emshaw | 77.05 | 3407 Stoughton Road | 9162541042 | CA | Sacramento | 2 |
| 5 | 229054573 | Clarabelle | Tearle | 82.16 | 69902 Dorton Junction | 7166276139 | NY | Buffalo | 2 |
| 6 | 251747956 | Aubine | Bemdtsson | 85.97 | 250 Linden Trail | 5023814520 | KY | Louisville | 2 |
| 7 | 2627959571 | Susi | Sharvill | 37.13 | 466 Sundown Pass | 9167692822 | CA | Sacramento | 3 |
| 8 | 2985053385 | Danica | Kensley | 88.88 | 788 Havey Court | 6261594582 | CA | Alhambra | 3 |
| 9 | 3492369222 | Winifield | Winskill | 47.14 | 97 3rd Pass | 5152085062 | IA | Des Moines | 4 |
| 10 | 362079200 | York | Curgenuer | 88.84 | 486 Lake View Terrace | 5402905768 | VA | Roanoke | 4 |
| 11 | 4063789551 | Albrecht | Marcum | 71.09 | 971 Oakridge Drive | 3109325900 | CA | Santa Monica | 4 |
| 12 | 4183901477 | Ker | Rebbeck | 77.43 | 0088 Wamer Drive | 7182926228 | NY | New York City | 5 |
| 13 | 4634033099 | Emanuel | Comelius | 91.92 | 51 Little Fleur Park | 3133227214 | MI | Detroit | 5 |
| 14 | 52765873 | Larina | Ritter | 57.68 | 817 McGuire Point | 3035498712 | CO | Denver | 5 |
| 15 | 5436655387 | Caye | Hamnett | 24.48 | 9217 Dixon Junction | 5124385325 | TX | Austin | 5 |
| 16 | 5477528289 | Durante | Dreier | 76.82 | 73 Warbler Drive | 5134194849 | OH | Cincinnati | 6 |
| 17 | 572549237 | Ree | Pauler | 54.52 | 63971 Ohio Parkway | 6031494909 | NH | Manchester | 6 |
| 18 | 5804237177 | Leola | Gannan | 22.16 | 05731 Lindbergh Junction | 2109384822 | TX | San Antonio | 6 |
| 19 | 5840844404 | Nanette | Girardengo | 19.10 | 2150 Surrey Parkway | 3613762359 | TX | Corpus Christi | 6 |
| 20 | 5864813592 | Rorie | Betancourt | 33.58 | 27440 Caliangt Way | 4154904618 | CA | San Francisco | 6 |
| 21 | 5966772740 | Kim | Gagen | 99.60 | 7012 Cambridge Parkway | 7279903384 | FL | Clearwater | 7 |
| 22 | 6011962951 | Aryt | Carlsson | 46.69 | 4273 Jenna Point | 5139756243 | OH | Cincinnati | 7 |
| 23 | 6626125260 | Ilyssa | Mold | 59.62 | 67741 Gateway Parkway | 4804043193 | AZ | Phoenix | 8 |
| 24 | 6918216884 | Gabbie | Asquith | 38.68 | 040 Eagan Trail | 3129173112 | IL | Chicago | 8 |
| 25 | 6975164124 | Rob | Rolston | 83.64 | 8 Arkansas Lane | 5611606254 | FL | Boca Raton | 8 |
| 26 | 7583088441 | Gilberta | Way | 91.38 | 0094 Fair Oaks Pass | 3219209624 | FL | Melbourne | 8 |
| 27 | 7601577954 | Dillon | Jillett | 86.55 | 9 Darwin Lane | 2601125755 | IN | Fort Wayne | 8 |
| 28 | 7895046015 | Dannel | Oyley | 86.13 | 1372 Warior Circle | 2025327659 | DC | Washington | 9 |
| 29 | 8187560116 | Shemie | McKiddin | 85.79 | 6122 Bonner Street | 6264824160 | CA | Los Angeles | 9 |
| 30 | 929024070 | Roshelle | Simond | 58.03 | 4 Eggendart Circle | 9175274534 | NY | Jamaica | 9 |
| 31 | 977324890 | Sheffy | Hinkley | 73.26 | 20961 Aberg Way | 2062857615 | WA | Seattle | 1 |
| 32 | 9918294620 | Rollin | Rollins | 81.27 | 38831 Michigan Lane | 7132344169 | TX | Humble | 10 |

SERVICES Table

| | ServiceID | ServiceDesc | ServiceRate |
|----|-----------|---|-------------|
| 1 | 1 | Standard Inspection (Interior) | 350.00 |
| 2 | 2 | Standard Inspection (Exterior) | 350.00 |
| 3 | 3 | Standard Inspection (Interior and Exterior) | 600.00 |
| 4 | 4 | On-Site Delivery Acceptance | 75.00 |
| 5 | 5 | On-Site Installation | 150.00 |
| 6 | 6 | Housekeeping | 150.00 |
| 7 | 7 | Pest Control | 60.00 |
| 8 | 8 | Pool Service | 90.00 |
| 9 | 9 | Landscaping | 200.00 |
| 10 | 10 | Plumbing | 100.00 |

CLIENTS Table

| | ClientID | CIAddress | CIPhone | YearsServiced | CIEmail | CIState | CICity | CFName | CLName | CIZip |
|----|----------|---------------------------|------------|---------------|-------------------------------|---------|----------------|------------|-----------|-------|
| 1 | 18503 | 1725 Slough Ave | 3141592653 | 7 | bestboss@dundermifflin.com | PA | Scranton | Michael | Scott | 18503 |
| 2 | 1000000 | 22 Red Ave | 4159574029 | 17 | tswizzle@weartshirts.com | CA | San Francisco | Taylor | Swift | 90021 |
| 3 | 1234567 | 111 West Ave | 1234567890 | 19 | pleasefail@test.com | TX | Austin | Please | Fail | 12345 |
| 4 | 2738192 | 123 South 5th St | 1827762154 | 2 | djskdie@gmail.com | AZ | Tempe | Walmart | NULL | 41296 |
| 5 | 3496507 | 4 Loomis Point | 4159574029 | 17 | vjakubowicz@howstuffworks.com | CA | San Francisco | Carlo | Hansen | 41283 |
| 6 | 3548957 | 9378 Sugar Way | 6024770379 | 16 | zspanswickn@twitpic.com | AZ | Phoenix | Carol | Vasquez | 88462 |
| 7 | 4812308 | 148 Transport Parkway | 2146294515 | 19 | cdunstans@wikipedia.org | TX | Dallas | Efran | Byrd | 69378 |
| 8 | 5000000 | 123 Main St | 5555555555 | 1 | test@test.com | AZ | Phoenix | Test | Subject | 10000 |
| 9 | 6586146 | 8 Kim Pass | 2813112153 | 2 | measonu@shareasale.com | TX | Houston | Carolin | Johnson | 40226 |
| 10 | 7101753 | 4838 Lakeland Terrace | 7245078034 | 5 | cskatcher9@behance.net | PA | New Castle | Elton | Johnson | 74842 |
| 11 | 8565248 | 35 Tony Avenue | 2127324383 | 12 | mshellum2@marketwatch.com | NY | New York City | Tyrone | Macias | 50822 |
| 12 | 8977331 | 17912 Farmco Parkway | 5097900342 | 23 | hswabyd@cdc.gov | WA | Spokane | Beverlee | Castaneda | 98088 |
| 13 | 9038853 | 936 Amoth Crossing | 7042993362 | 23 | sstur3@tumblr.com | NC | Charlotte | Alena | Klein | 11865 |
| 14 | 18896384 | 266 Weeping Birch Circle | 3121237328 | 17 | smurthaitem@blinklist.com | IL | Chicago | Rigoberto | Finley | 29529 |
| 15 | 20536872 | 441 South Avenue | 7162972323 | 24 | mcantero@loc.gov | NY | Buffalo | Lucius | Bowers | 72504 |
| 16 | 36328797 | 2946 West Lane | 4058676001 | 23 | salesb@wikispaces.com | OK | Oklahoma City | Shirly | Beck | 85409 |
| 17 | 39862515 | 1220 8th Plaza | 9379393341 | 15 | sboshardq@wikispaces.com | OH | Springfield | Charisse | Braun | 85651 |
| 18 | 45730150 | 453 Miller Crossing | 7065809309 | 20 | lmewitta@meetup.com | GA | Cumming | Sparkle | Graham | 12369 |
| 19 | 47620087 | 15 Walton Pass | 860899058 | 13 | hrenton0@ted.com | CT | Hartford | Modesto | Oneal | 47693 |
| 20 | 50655329 | 2979 Sundown Terrace | 4056290290 | 14 | sjulyanr@howstuffworks.com | OK | Oklahoma City | Kenyatta | Carroll | 87108 |
| 21 | 55034572 | 1 Dexter Crossing | 8603854839 | 20 | epudan7@example.com | CT | Hartford | Kristopher | Dougherty | 33292 |
| 22 | 61406868 | 2012 Blackbird Junction | 8059920105 | 22 | pbehnecke1@zdnet.com | CA | Bakersfield | Fredricka | Williams | 15147 |
| 23 | 64593989 | 17 Fulton Road | 3257507833 | 7 | tgoodanewf@networkchange.org | TX | San Angelo | Maxie | Daugherty | 43526 |
| 24 | 65321745 | 75 Amoth Avenue | 2036971420 | 14 | bjandeinh@cnn.com | CT | New Haven | Laurene | Heming | 73131 |
| 25 | 66233683 | 902 Lotheville Road | 5131931977 | 21 | tgodlontone@networkchange.org | OH | Cincinnati | Brett | Randall | 62606 |
| 26 | 68203269 | 229 David Point | 2029142772 | 22 | rbainesv@npr.org | DC | Washington | Erika | Andrews | 99195 |
| 27 | 68603197 | 5 Brown Trail | 3177883221 | 6 | mcchiommiert15@usgs.gov | IN | Indianapolis | Vernon | Cannon | 55876 |
| 28 | 79147483 | 38294 Maple Parkway | 6096763322 | 20 | jhaythi@imageshack.us | NJ | Trenton | Cary | Duarte | 81876 |
| 29 | 80920878 | 57458 American Ash Street | 5208881919 | 13 | kmowsdill@phoca.cz | AZ | Tucson | Charisse | Braun | 71734 |
| 30 | 81164013 | 53675 Alpine Street | 8318228004 | 13 | rexposito4@ucoz.com | CA | Salinas | Arianne | Foster | 36945 |
| 31 | 83443310 | 7762 Cottonwood Point | 9164703118 | 4 | pwallbutton1@de.vu | CA | Sacramento | Dodie | Ford | 13572 |
| 32 | 83696889 | 09320 Hansons Place | 7868831170 | 11 | ghaughton8@ebay.com | FL | Miami | Ahmed | Wade | 64647 |
| 33 | 87915581 | 26 Lakeland Way | 7164958639 | 9 | ljorczykt@pcworld.com | NY | Buffalo | Collette | Clements | 81660 |
| 34 | 91761429 | 3 Myrtle Park | 7192665655 | 7 | lstrowan6@odnoklassniki.ru | CO | Colorado Sp... | Cody | Delacruz | 11561 |
| 35 | 93971647 | 4 Eagan Avenue | 7049563884 | 9 | yohareg@fda.gov | NC | Charlotte | Madelai... | Dunn | 70399 |
| 36 | 95264136 | 1 Rieder Crossing | 2125065536 | 15 | sweatherburnk@tripod.com | NY | New York City | Elinor | Gay | 12786 |
| 37 | 95811088 | 5 Mitchell Park | 8586468390 | 12 | ewalasikj@1688.com | CA | San Diego | Rodrigo | Melton | 63538 |

PROPERTIES Table

| | PropertyID | PropName | Address | #Bed | #Bath | SqFt | YrBuilt | RoofType | ExteriorType | ClientID |
|----|------------|---------------|---------------------------|------|-------|---------|---------|----------|--------------|----------|
| 1 | 1000 | Subject | 123 Main St | 1 | 1 | 1100.00 | 2019 | tile | brick | 5000000 |
| 2 | 1001 | Bennett | 2946 West Lane | 3 | 3 | 2670.00 | 2006 | concrete | brick | 1000000 |
| 3 | 1106 | Johnson | 8 Kim Pass | 2 | 1 | 1080.00 | 2003 | tile | stucco | 6586146 |
| 4 | 1725 | DunderMifflin | 1725 Slough Avenue | 1 | 1 | 20.00 | 2000 | concrete | brick | 18503 |
| 5 | 1811 | Melton | 5 Mitchell Park | 3 | 2 | 2200.00 | 1990 | shingles | stucco | 95811088 |
| 6 | 2128 | Mcintyre | 57458 American Ash Street | 2 | 2 | 1500.00 | 1993 | shingles | stucco | 80920878 |
| 7 | 3016 | Klein | 936 Amoth Crossing | 3 | 3 | 2670.00 | 2007 | shingles | stucco | 9038853 |
| 8 | 3190 | Gay | 1 Rieder Crossing | 3 | 3 | 2670.00 | 2012 | concrete | vinyl | 95264136 |
| 9 | 4133 | Cannon | 5 Brown Trail | 3 | 2 | 2200.00 | 2008 | shingles | vinyl | 68603197 |
| 10 | 4233 | Williams | 2012 Blackbird Junction | 4 | 3 | 2985.00 | 2006 | concrete | vinyl | 61406868 |
| 11 | 4270 | Randall | 902 Lotheville Road | 8 | 5 | 6250.00 | 1997 | tile | stucco | 66233683 |
| 12 | 4630 | Dougherty | 1 Dexter Crossing | 3 | 2 | 1935.00 | 1998 | tile | vinyl | 55034572 |
| 13 | 4855 | Hering | 75 Amoth Avenue | 4 | 3 | 3500.00 | 2007 | tile | vinyl | 65321745 |
| 14 | 4879 | Dunn | 4 Eagan Avenue | 8 | 5 | 6250.00 | 2007 | tile | stucco | 93971647 |
| 15 | 5555 | Johnson | 4838 Lakeland Terrace | 2 | 1 | 1080.00 | 2013 | concrete | stucco | 7101753 |
| 16 | 5726 | Duarte | 38294 Maple Parkway | 3 | 2 | 2200.00 | 2010 | concrete | brick | 79147483 |
| 17 | 5754 | Bowers | 441 South Avenue | 4 | 3 | 2985.00 | 1998 | shingles | brick | 20536872 |
| 18 | 5882 | Vasquez | 9378 Sugar Way | 6 | 4 | 4520.00 | 2006 | concrete | stucco | 3548957 |
| 19 | 6434 | Delacruz | 3 Myrtle Park | 2 | 1 | 1080.00 | 1995 | tile | brick | 91761429 |
| 20 | 6717 | Byrd | 148 Transport Parkway | 1 | 1 | 525.00 | 2001 | concrete | brick | 4812308 |
| 21 | 6789 | Andrews | 229 David Point | 2 | 1 | 1050.00 | 2007 | shingles | vinyl | 68203269 |
| 22 | 6929 | Oneal | 15 Walton Pass | 3 | 2 | 1935.00 | 2001 | tile | brick | 47620087 |
| 23 | 7059 | Carroll | 2979 Sundown Terrace | 4 | 3 | 2985.00 | 2005 | shingles | stucco | 50655329 |
| 24 | 7120 | Foster | 53675 Alpine Street | 3 | 3 | 2670.00 | 2005 | shingles | brick | 81164013 |
| 25 | 7388 | Hansen | 4 Loomis Point | 2 | 2 | 1500.00 | 2010 | tile | stucco | 3496507 |
| 26 | 7426 | Beck | 2946 West Lane | 4 | 3 | 2985.00 | 2006 | concrete | brick | 36328797 |
| 27 | 8000 | Macias | 35 Tony Avenue | 4 | 3 | 2985.00 | 1995 | concrete | stucco | 8565248 |
| 28 | 8123 | Castaneda | 17912 Famco Parkway | 4 | 3 | 2985.00 | 1998 | tile | vinyl | 8977331 |
| 29 | 8448 | Daugherty | 17 Fulton Road | 4 | 3 | 2980.00 | 2009 | shingles | stucco | 64593989 |
| 30 | 8770 | Ford | 7762 Cottonwood Point | 3 | 2 | 2200.00 | 1997 | concrete | brick | 83443310 |
| 31 | 8901 | Finley | 266 Weeping Birch Circle | 3 | 2 | 1935.00 | 2001 | concrete | stucco | 18896384 |
| 32 | 9051 | Braun | 1220 8th Plaza | 3 | 3 | 2670.00 | 2009 | tile | stucco | 39862515 |
| 33 | 9350 | Wade | 09320 Hansons Place | 3 | 2 | 1935.00 | 1999 | shingles | brick | 83696889 |
| 34 | 9355 | Graham | 453 Miller Crossing | 3 | 2 | 1935.00 | 1996 | tile | stucco | 45730150 |
| 35 | 9851 | Clements | 26 Lakeland Way | 2 | 1 | 1080.00 | 1999 | concrete | stucco | 87915581 |

VISITS Table

| | VisitID | DateOfVisit | StartDate | EndDate | Owner | TotalCost | PropertyID | ServiceID | ClientID | EmployeeID |
|----|---------|-------------|------------|------------|----------------------|-----------|------------|-----------|----------|------------|
| 1 | 4761 | 2017-04-27 | 2017-06-09 | 2019-03-03 | Brett Randall | 61540.12 | 4270 | 1 | 66233683 | 3492369222 |
| 2 | 8453 | 2017-02-25 | 2017-08-29 | 2018-04-22 | Modesto Oneal | 65027.39 | 6929 | 1 | 47620087 | 4634033099 |
| 3 | 10384 | 2017-01-15 | 2017-11-05 | 2019-07-18 | Lucius Bowers | 90846.03 | 5754 | 1 | 20536872 | 2627959571 |
| 4 | 26217 | 2017-02-25 | 2017-05-08 | 2018-08-26 | Sparkle Graham | 82603.56 | 9355 | 9 | 45730150 | 929024070 |
| 5 | 28319 | 2017-01-31 | 2017-04-29 | 2020-02-01 | Tyrone Macias | 32810.50 | 8000 | 1 | 8565248 | 1303503027 |
| 6 | 54238 | 2017-02-14 | 2017-10-30 | 2018-06-25 | Shirly Beck | 42291.77 | 7426 | 3 | 36328797 | 4183901477 |
| 7 | 130047 | 2017-05-09 | 2017-11-16 | 2019-02-11 | Rigoberto Finley | 52230.82 | 8901 | 6 | 18896384 | 4063789551 |
| 8 | 150336 | 2017-04-01 | 2017-12-01 | 2019-01-25 | Fredricka Williams | 90087.57 | 4233 | 1 | 61406868 | 7601577954 |
| 9 | 189333 | 2017-01-08 | 2017-03-22 | 2019-11-14 | Kristopher Dougherty | 91817.33 | 4630 | 1 | 55034572 | 5436655387 |
| 10 | 241683 | 2017-01-09 | 2017-07-21 | 2018-05-24 | Matthew McIntyre | 28903.56 | 2128 | 1 | 80920878 | 6975164124 |
| 11 | 266008 | 2017-05-17 | 2017-04-16 | 2018-09-03 | Laurene Herring | 74936.77 | 4855 | 5 | 65321745 | 229054573 |
| 12 | 350883 | 2017-02-02 | 2017-06-17 | 2020-02-08 | Efren Byrd | 48206.24 | 6717 | 3 | 4812308 | 7583088441 |
| 13 | 388938 | 2017-05-23 | 2017-04-25 | 2019-06-08 | Rodrigo Melton | 44289.93 | 1811 | 4 | 95811088 | 8187560116 |
| 14 | 391378 | 2017-02-14 | 2017-05-20 | 2019-06-17 | Dodie Ford | 97281.34 | 8770 | 7 | 83443310 | 5840844404 |
| 15 | 401071 | 2017-05-16 | 2017-10-28 | 2018-07-02 | Cody Delacruz | 42274.98 | 6434 | 6 | 91761429 | 362079200 |
| 16 | 417991 | 2017-03-23 | 2017-02-12 | 2018-12-18 | Elinor Gay | 37963.65 | 3190 | 2 | 95264136 | 5477528289 |
| 17 | 489302 | 2017-04-11 | 2017-08-28 | 2018-09-02 | Alena Klein | 22803.75 | 3016 | 4 | 9038853 | 52765873 |
| 18 | 501481 | 2017-04-06 | 2017-09-20 | 2018-09-09 | Charisse Braun | 46830.73 | 9051 | 9 | 39862515 | 251747956 |
| 19 | 506920 | 2017-03-23 | 2017-04-29 | 2018-05-25 | Ahmed Wade | 66515.82 | 9350 | 1 | 83696889 | 2985053385 |
| 20 | 627588 | 2017-01-26 | 2017-10-07 | 2020-02-25 | Kenyatta Carroll | 40056.08 | 7059 | 7 | 50655329 | 7895046015 |
| 21 | 681771 | 2017-01-16 | 2017-02-14 | 2018-11-20 | Collette Clements | 77142.88 | 9851 | 2 | 87915581 | 6918216884 |
| 22 | 752306 | 2017-05-18 | 2017-06-07 | 2018-03-29 | Vemon Cannon | 49014.88 | 4133 | 8 | 68603197 | 6011962951 |
| 23 | 773510 | 2017-05-29 | 2017-10-08 | 2019-09-03 | Maxie Daugherty | 34098.94 | 8448 | 1 | 64593989 | 5804237177 |
| 24 | 799045 | 2017-01-02 | 2017-06-09 | 2018-12-09 | Cary Duarte | 44753.40 | 5726 | 3 | 79147483 | 5864813592 |
| 25 | 833698 | 2017-03-27 | 2017-10-11 | 2018-01-08 | Beverlee Castaneda | 59993.24 | 8123 | 1 | 8977331 | 6626125260 |
| 26 | 840222 | 2017-02-11 | 2017-01-21 | 2018-09-17 | Carol Vasquez | 86354.72 | 5882 | 2 | 3548957 | 2075868651 |
| 27 | 930329 | 2017-03-17 | 2017-02-09 | 2018-10-14 | Elton Johnson | 90602.00 | 5555 | 5 | 7101753 | 9918294620 |
| 28 | 937667 | 2017-04-08 | 2017-11-03 | 2019-12-31 | Madelaine Dunn | 97975.29 | 4879 | 1 | 93971647 | 977324890 |
| 29 | 986315 | 2017-05-22 | 2017-10-14 | 2019-03-13 | Arianne Foster | 26578.87 | 7120 | 2 | 81164013 | 572549237 |
| 30 | 996620 | 2017-05-15 | 2017-12-15 | 2019-04-20 | Carlo Hansen | 27828.25 | 7388 | 2 | 3496507 | 5966772740 |
| 31 | 999000 | 2020-05-01 | 2020-06-06 | 2020-06-13 | Michael Scott | 999.00 | 1725 | 10 | 18503 | 111111 |

APPOINTMENTS Table

Too long for one page; split into two columns.

| | AppointmentID | VisitID | ServiceID | EmployeeID | | | |
|----|---------------|---------|-----------|------------|----|--------|--------|
| 1 | 5159 | 501481 | 9 | 2627959571 | 31 | 445774 | 391378 |
| 2 | 17992 | 130047 | 6 | 5804237177 | 32 | 467918 | 501481 |
| 3 | 43058 | 799045 | 3 | 977324890 | 33 | 552000 | 150336 |
| 4 | 48216 | 130047 | 6 | 5477528289 | 34 | 567939 | 937667 |
| 5 | 51161 | 266008 | 5 | 229054573 | 35 | 583052 | 417991 |
| 6 | 60439 | 937667 | 1 | 7583088441 | 36 | 586152 | 773510 |
| 7 | 62077 | 391378 | 7 | 3492369222 | 37 | 617200 | 388938 |
| 8 | 77719 | 752306 | 8 | 52765873 | 38 | 638983 | 937667 |
| 9 | 78245 | 506920 | 1 | 362079200 | 39 | 664484 | 266008 |
| 10 | 112013 | 930329 | 5 | 5436655387 | 40 | 690337 | 996620 |
| 11 | 112384 | 26217 | 9 | 2985053385 | 41 | 702561 | 391378 |
| 12 | 130592 | 996620 | 2 | 572549237 | 42 | 705045 | 350883 |
| 13 | 131115 | 10384 | 1 | 362079200 | 43 | 707491 | 350883 |
| 14 | 140617 | 489302 | 4 | 7601577954 | 44 | 760380 | 986315 |
| 15 | 165645 | 833698 | 1 | 6011962951 | 45 | 769504 | 8453 |
| 16 | 191304 | 840222 | 2 | 6011962951 | 46 | 779760 | 773510 |
| 17 | 199047 | 28319 | 1 | 4634033099 | 47 | 802776 | 681771 |
| 18 | 210221 | 999000 | 6 | 5804237177 | 48 | 840011 | 417991 |
| 19 | 271937 | 4761 | 1 | 5966772740 | 49 | 856578 | 388938 |
| 20 | 281579 | 54238 | 3 | 52765873 | 50 | 867476 | 833698 |
| 21 | 293772 | 986315 | 2 | 1303503027 | 51 | 867845 | 266008 |
| 22 | 296873 | 189333 | 1 | 6011962951 | 52 | 869979 | 26217 |
| 23 | 320555 | 840222 | 2 | 5840844404 | 53 | 881980 | 388938 |
| 24 | 390382 | 241683 | 1 | 2075868651 | 54 | 922622 | 130047 |
| 25 | 393515 | 10384 | 1 | 4063789551 | 55 | 932009 | 54238 |
| 26 | 397679 | 799045 | 3 | 4183901477 | 56 | 935619 | 930329 |
| 27 | 412300 | 417991 | 2 | 1303503027 | 57 | 937753 | 773510 |
| 28 | 422144 | 627588 | 7 | 6975164124 | 58 | 977179 | 4761 |
| 29 | 429189 | 627588 | 7 | 9918294620 | 59 | 981999 | 401071 |
| 30 | 435366 | 840222 | 2 | 977324890 | 60 | 982579 | 241683 |
| | | | | | 61 | 985680 | 799045 |

CORPORATE Table

| | ContactName | ContactPhone | ClientID |
|----|-------------------|--------------|----------|
| 1 | Raddie Velareal | 7868105160 | 9038853 |
| 2 | Reggie Renton | 9155973136 | 81164013 |
| 3 | Lynelle Frowde | 8624397881 | 68603197 |
| 4 | Dave Linthead | 8506516447 | 91761429 |
| 5 | Roger Canet | 7045556346 | 55034572 |
| 6 | Bev Buras | 9187985896 | 83696889 |
| 7 | Katalin Ladbury | 4083015743 | 7101753 |
| 8 | Bronnie Chemy | 4045406276 | 45730150 |
| 9 | Betteann Braunton | 8634429086 | 36328797 |
| 10 | Hobie Bumby | 5638956957 | 3496507 |
| 11 | Alfonse Ebhardt | 5858746191 | 8977331 |
| 12 | Linet Temis | 6081403315 | 20536872 |
| 13 | Jillene Penna | 3145806106 | 80920878 |
| 14 | Sharyl Widdop | 6153672182 | 39862515 |
| 15 | Jamil Borkett | 4132574932 | 50655329 |

INDIVIDUAL Table

| | CreditCard | ClientID |
|----|-----------------|----------|
| 1 | 392465468562285 | 47620087 |
| 2 | 737875881691977 | 61406868 |
| 3 | 915428977813510 | 8565248 |
| 4 | 205430679934836 | 66233683 |
| 5 | 452468253896930 | 64593989 |
| 6 | 995621911597674 | 93971647 |
| 7 | 787617247129312 | 65321745 |
| 8 | 794980363179730 | 79147483 |
| 9 | 466604156551278 | 95811088 |
| 10 | 224623828193391 | 95264136 |
| 11 | 431158481165622 | 83443310 |
| 12 | 416814892773282 | 18896384 |
| 13 | 523083297614496 | 3548957 |
| 14 | 350101724221769 | 4812308 |
| 15 | 933813096207303 | 87915581 |

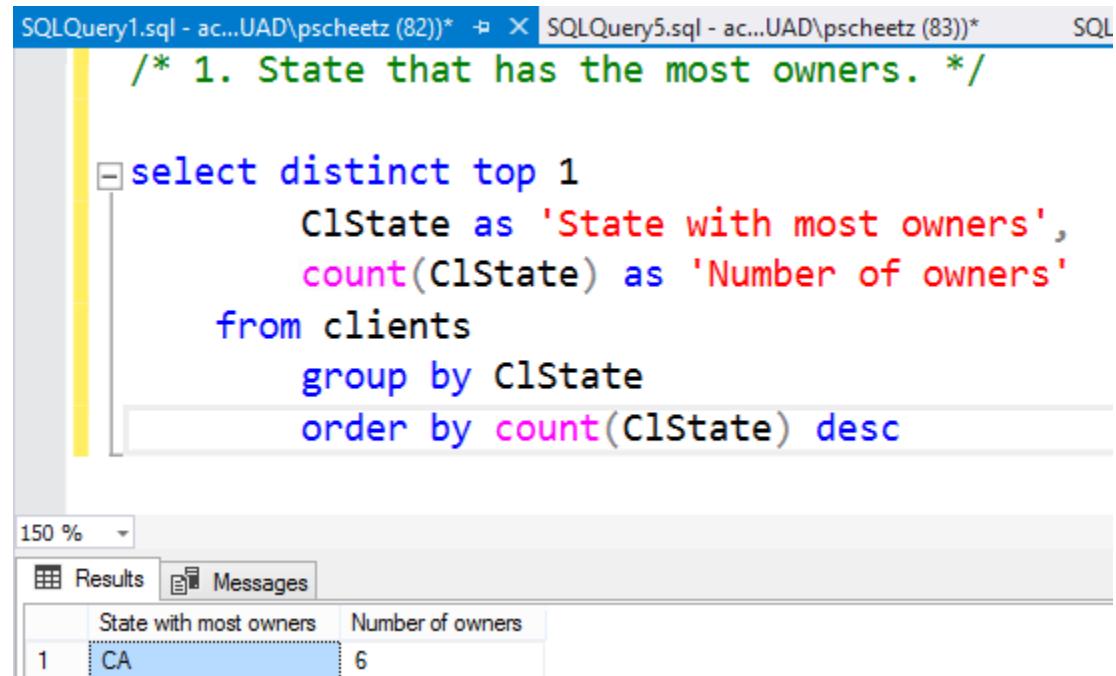
SERVICECOMPLETION Table

| | VisitID | Season | StartDate | EndDate | DaysToFin | PropertyID | EFName | ELName | Specialization |
|---|---------|--------|------------|------------|-----------|------------|----------|--------|----------------|
| 1 | 26217 | Summer | 2017-05-08 | 2018-08-26 | 475 | 9355 | Roshelle | Simond | 5 |

Screenshots of all Queries/Results

Client Requested Queries

1. State that has the most owners.



The screenshot shows a SQL Server Management Studio (SSMS) interface. At the top, there are two tabs: 'SQLQuery1.sql - ac...UAD\pscheetz (82))*' and 'SQLQuery5.sql - ac...UAD\pscheetz (83))*'. The main area contains a query script:

```
/* 1. State that has the most owners. */

select distinct top 1
    C1State as 'State with most owners',
    count(C1State) as 'Number of owners'
from clients
group by C1State
order by count(C1State) desc
```

Below the script, there is a results pane with a grid. The grid has two columns: 'State with most owners' and 'Number of owners'. A single row is shown, with the value 'CA' highlighted in blue:

| | State with most owners | Number of owners |
|---|------------------------|------------------|
| 1 | CA | 6 |

2. Number/list of owners in each state.

```

SQLQuery1.sql - ac...UAD\pscheetz (82)* ➔ SQLQuery5.sql - ac...UAD\pscheetz (83)* ➔ SQLQuery3.sql - ac...UAD\pscheetz (69)*
/* 2. Number/list of owners in each state. */

select ClientID, ClState as 'State',
       CFName as 'First Name', CLname as 'Last Name'
  from CLIENTS
 order by ClState, CFName asc;

select distinct
       ClState as 'State',
       count(ClState) as 'Number of owners'
  from clients
 group by ClState
 order by count(ClState) desc;

```

| | ClientID | State | First Name | Last Name |
|----|----------|-------|------------|-----------|
| 1 | 3548957 | AZ | Carol | Vasquez |
| 2 | 80920878 | AZ | Charisse | Braun |
| 3 | 5000000 | AZ | Test | Subject |
| 4 | 2738192 | AZ | Walmart | NULL |
| 5 | 81164013 | CA | Arianne | Foster |
| 6 | 3496507 | CA | Carlo | Hansen |
| 7 | 83443310 | CA | Dodie | Ford |
| 8 | 61406868 | CA | Fredricka | Williams |
| 9 | 95811088 | CA | Rodrigo | Melton |
| 10 | 1000000 | CA | Taylor | Swift |
| 11 | 91761429 | CO | Cody | Delacruz |
| 12 | 55034572 | CT | Kristopher | Dougherty |
| 13 | 65321745 | CT | Laurene | Herring |
| 14 | 47620087 | CT | Modesto | Oneal |
| 15 | 68203269 | DC | Erika | Andrews |
| 16 | 83696889 | FL | Ahmed | Wade |
| 17 | 45730150 | GA | Sparkle | Graham |
| 18 | 18896384 | IL | Rigoberto | Finley |
| 19 | 68603197 | IN | Vernon | Cannon |
| 20 | 9038853 | NC | Alena | Klein |
| 21 | 93971647 | NC | Madelaine | Dunn |
| 22 | 79147483 | NJ | Cary | Duarte |
| 23 | 87915581 | NY | Collette | Clements |
| 24 | 95264136 | NY | Elinor | Gay |
| 25 | 20536872 | NY | Lucius | Bowers |
| 26 | 8565248 | NY | Tyrone | Macias |
| 27 | 66233683 | OH | Brett | Randall |
| 28 | 39862515 | OH | Charisse | Braun |
| 29 | 50655329 | OK | Kenyatta | Carroll |
| 30 | 36328797 | OK | Shirly | Beck |
| 31 | 7101753 | PA | Elton | Johnson |
| 32 | 18503 | PA | Michael | Scott |
| 33 | 6586146 | TX | Carolin | Johnson |
| 34 | 4812308 | TX | Efren | Byrd |
| 35 | 64593989 | TX | Maxie | Daugherty |
| 36 | 1234567 | TX | Please | Fai |
| 37 | 8977331 | WA | Beverlee | Castaneda |

| | State | Number of owners |
|----|-------|------------------|
| 1 | CA | 6 |
| 2 | NY | 4 |
| 3 | TX | 4 |
| 4 | AZ | 4 |
| 5 | CT | 3 |
| 6 | NC | 2 |
| 7 | OH | 2 |
| 8 | OK | 2 |
| 9 | PA | 2 |
| 10 | WA | 1 |
| 11 | NJ | 1 |
| 12 | CO | 1 |
| 13 | DC | 1 |
| 14 | FL | 1 |
| 15 | GA | 1 |
| 16 | IL | 1 |
| 17 | IN | 1 |

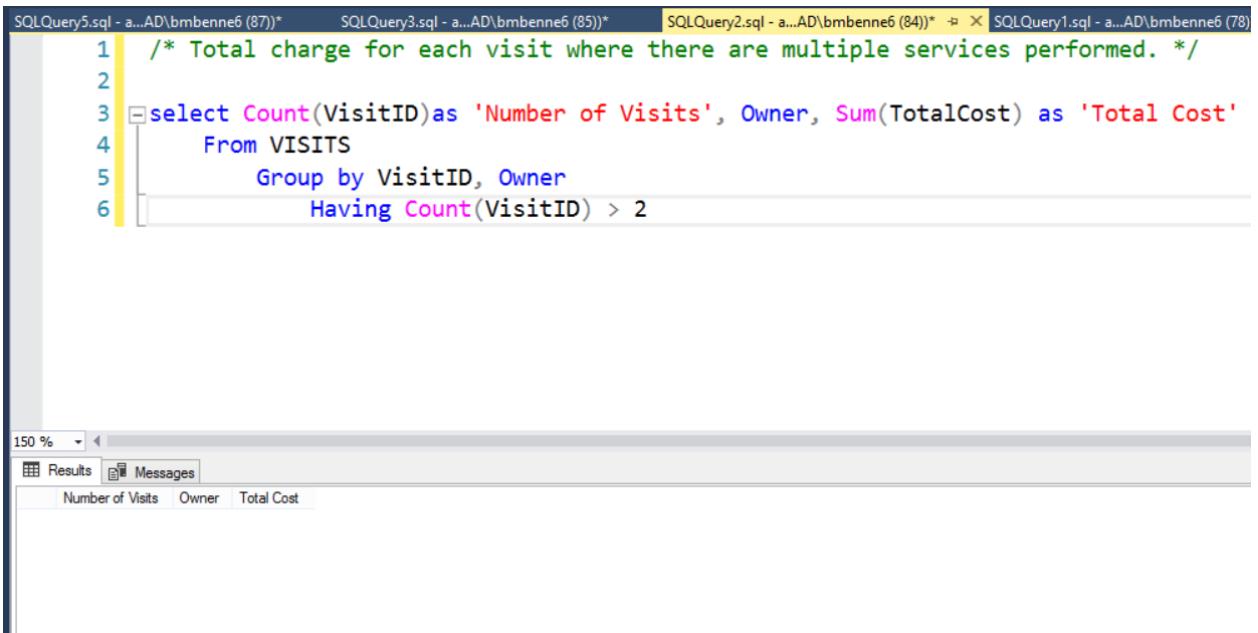
3. Total number of visits for each client for each season.

```
1 /* Total number of visits for each client for each season. */
2
3 Select CFName, CLName, Count(VisitID) as 'Number of Visits',
4      (case when month(DateOfVisit) in (12, 1, 2) then 'Winter'
5           when month(DateOfVisit) in (3, 4, 5) then 'Spring'
6           when month(DateOfVisit) in (6, 7, 8) then 'Summer'
7           when month(DateOfVisit) in (9, 10, 11) then 'Autumn'
8           end) as 'Season'
9      From CLIENTS C
10     JOIN VISITS V
11       on C.ClientID = V.ClientID
12      Group By CFName, CLName, DateOfVisit
```

Results

| | CFName | CLName | Number of Visits | Season |
|----|----------|-----------|------------------|--------|
| 1 | Ahmed | Wade | 1 | Spring |
| 2 | Alena | Klein | 1 | Spring |
| 3 | Arianne | Foster | 1 | Spring |
| 4 | Beverlee | Castaneda | 1 | Spring |
| 5 | Brett | Randall | 1 | Spring |
| 6 | Carlo | Hansen | 1 | Spring |
| 7 | Carol | Vasquez | 1 | Winter |
| 8 | Cary | Duarte | 1 | Winter |
| 9 | Charisse | Braun | 1 | Winter |
| 10 | Charisse | Braun | 1 | Spring |
| 11 | Cody | Delacruz | 1 | Spring |
| 12 | Collette | Clements | 1 | Winter |
| 13 | Dodie | Ford | 1 | Winter |

4. Total charge for each visit where there are multiple services performed.



The screenshot shows a SQL Server Management Studio interface. At the top, there are four tabs: 'SQLQuery5.sql - a...AD\bmbenne6 (87)*', 'SQLQuery3.sql - a...AD\bmbenne6 (85)*', 'SQLQuery2.sql - a...AD\bmbenne6 (84)*', and 'SQLQuery1.sql - a...AD\bmbenne6 (78)'. The fourth tab is active. Below the tabs, a code editor displays the following SQL query:

```
1 /* Total charge for each visit where there are multiple services performed. */
2
3 select Count(VisitID)as 'Number of Visits', Owner, Sum(TotalCost) as 'Total Cost'
4     From VISITS
5         Group by VisitID, Owner
6             Having Count(VisitID) > 2
```

Below the code editor is a results pane with a 'Results' tab selected. The results table has three columns: 'Number of Visits', 'Owner', and 'Total Cost'. There are no rows displayed in the results table.

**** NOTE -- There are no instances where a customer had more than one visit**

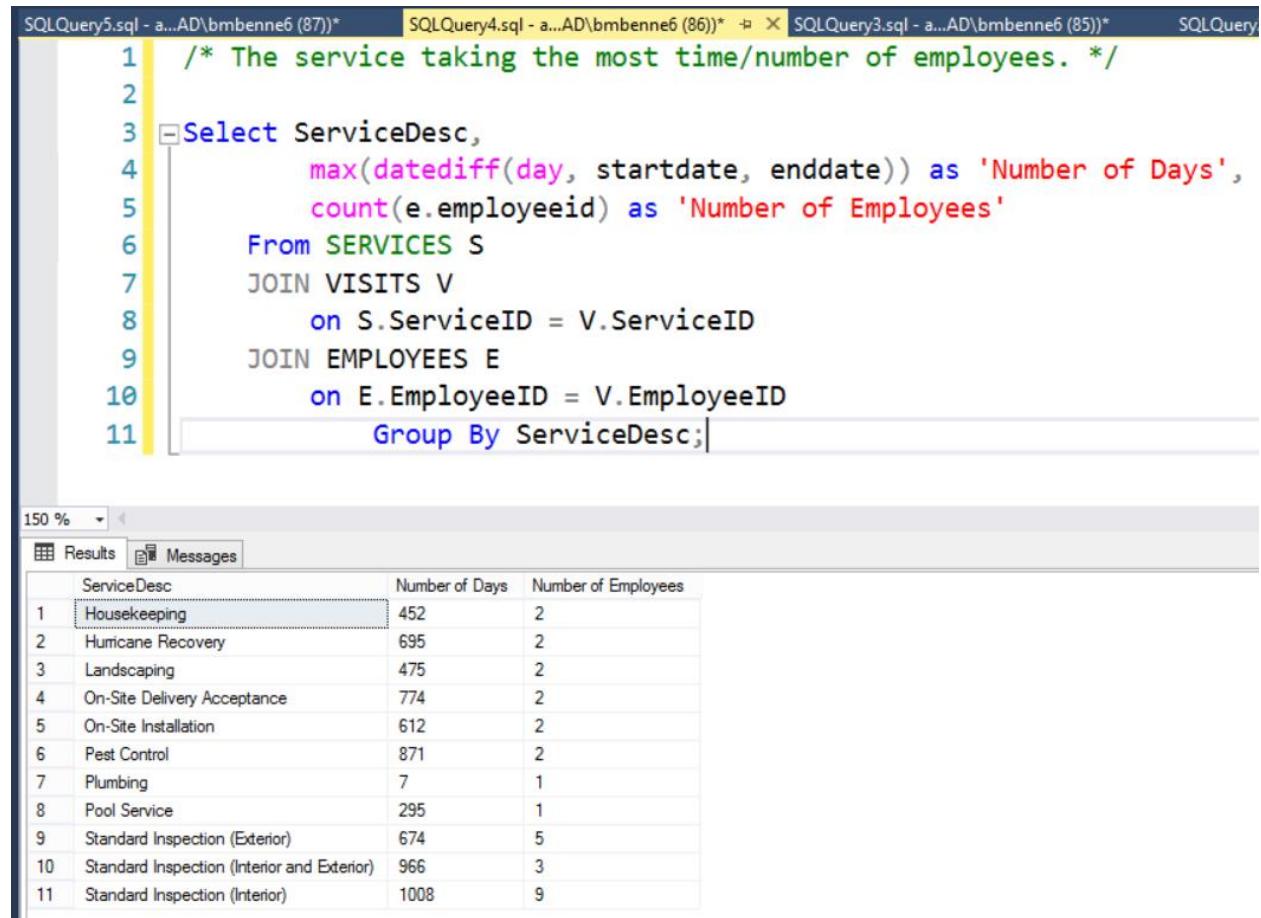
5. A list of clients who have ordered pest control.

```
SQLQuery5.sql - a...AD\bmbenne6 (87)*      SQLQuery4.sql - a...AD\bmbenne6 (86)*      SQLQuery3.sql - a...AD\bmbenne6 (85)*
1  /* 5. A list of clients who have ordered pest control. */
2
3  Select C.ClientID, CFName, CLName, ServiceDesc
4    From Clients C
5    JOIN VISITS V
6      on C.ClientID = V.ClientID
7    JOIN SERVICES S
8      on V.ServiceID = S.ServiceID
9    Where ServiceDesc = 'Pest Control'
```

150 %

| | ClientID | CFName | CLName | ServiceDesc |
|---|----------|----------|---------|--------------|
| 1 | 83443310 | Dodie | Ford | Pest Control |
| 2 | 50655329 | Kenyatta | Carroll | Pest Control |

6. The service taking the most time/number of employees.



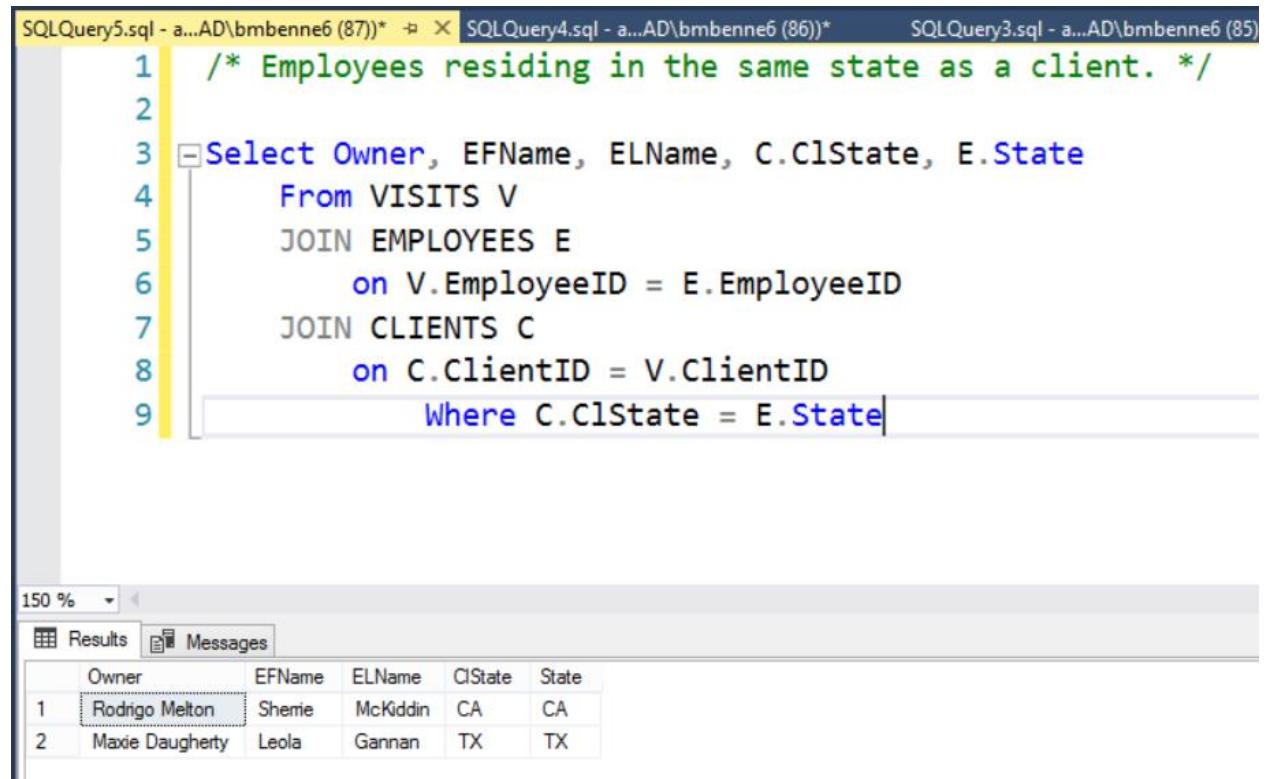
The screenshot shows a SQL Server Management Studio window with multiple tabs at the top. The active tab is 'SQLQuery5.sql - a...AD\bmbenne6 (87)*'. Below the tabs is a code editor containing the following SQL query:

```
1  /* The service taking the most time/number of employees. */
2
3  Select ServiceDesc,
4      max(datediff(day, startdate, enddate)) as 'Number of Days',
5      count(e.employeeid) as 'Number of Employees'
6  From SERVICES S
7  JOIN VISITS V
8      on S.ServiceID = V.ServiceID
9  JOIN EMPLOYEES E
10     on E.EmployeeID = V.EmployeeID
11     Group By ServiceDesc;
```

Below the code editor is a results grid titled 'Results' with the following data:

| | ServiceDesc | Number of Days | Number of Employees |
|----|---|----------------|---------------------|
| 1 | Housekeeping | 452 | 2 |
| 2 | Hurricane Recovery | 695 | 2 |
| 3 | Landscaping | 475 | 2 |
| 4 | On-Site Delivery Acceptance | 774 | 2 |
| 5 | On-Site Installation | 612 | 2 |
| 6 | Pest Control | 871 | 2 |
| 7 | Plumbing | 7 | 1 |
| 8 | Pool Service | 295 | 1 |
| 9 | Standard Inspection (Exterior) | 674 | 5 |
| 10 | Standard Inspection (Interior and Exterior) | 966 | 3 |
| 11 | Standard Inspection (Interior) | 1008 | 9 |

7. Employees residing in the same state as a client.



The screenshot shows a SQL Server Management Studio window with three tabs at the top: 'SQLQuery5.sql - a...AD\bmbenne6 (87)*' (active), 'SQLQuery4.sql - a...AD\bmbenne6 (86)*', and 'SQLQuery3.sql - a...AD\bmbenne6 (85)'. The main area contains a SQL query:

```
1  /* Employees residing in the same state as a client. */
2
3  Select Owner, EFName, ELName, C.ClState, E.State
4    From VISITS V
5      JOIN EMPLOYEES E
6          on V.EmployeeID = E.EmployeeID
7      JOIN CLIENTS C
8          on C.ClientID = V.ClientID
9      Where C.ClState = E.State
```

Below the query, the results pane is visible, showing a table with five columns: Owner, EFName, ELName, ClState, and State. There are two rows of data:

| | Owner | EFName | ELName | ClState | State |
|---|-----------------|--------|----------|---------|-------|
| 1 | Rodrigo Melton | Sherie | McKiddin | CA | CA |
| 2 | Maxie Daugherty | Leola | Gannan | TX | TX |

8. List of employees performing the most services.

SQLQuery1.sql - a...UAD\bmdavi24 (56)* X

```
SELECT A.EmployeeID, EFName, ELName, COUNT(ServiceID) AS 'Number of Services'
FROM EMPLOYEES E
JOIN APPOINTMENTS A ON E.EmployeeID = A.EmployeeID
GROUP BY A.EmployeeID, EFName, ELName
ORDER BY COUNT(ServiceID) DESC;
```

100 % < >

Results Messages

| | EmployeeID | EFName | ELName | Number of Services |
|----|------------|-------------|-------------|--------------------|
| 1 | 5966772740 | Kim | Gagen | 4 |
| 2 | 6011962951 | Arty | Carlsson | 3 |
| 3 | 7583088441 | Gilberta | Way | 3 |
| 4 | 9918294620 | Rollin | Rollins | 3 |
| 5 | 1303503027 | Ulrikaumeko | De Dantesie | 3 |
| 6 | 229054573 | Clarabelle | Tearle | 3 |
| 7 | 2985053385 | Danica | Kensley | 3 |
| 8 | 4183901477 | Ker | Rebbeck | 3 |
| 9 | 52765873 | Larina | Ritter | 3 |
| 10 | 5436655387 | Caye | Hamnett | 3 |
| 11 | 572549237 | Ree | Pauler | 3 |
| 12 | 5804237177 | Leola | Gannan | 3 |
| 13 | 4634033099 | Emanuel | Comelius | 2 |
| 14 | 3492369222 | Winifield | Winskill | 2 |
| 15 | 362079200 | York | Curgenuer | 2 |
| 16 | 977324890 | Sheffy | Hinkley | 2 |
| 17 | 7601577954 | Dillon | Jillett | 2 |

Query executed successfully.

9. Total revenue received last year.

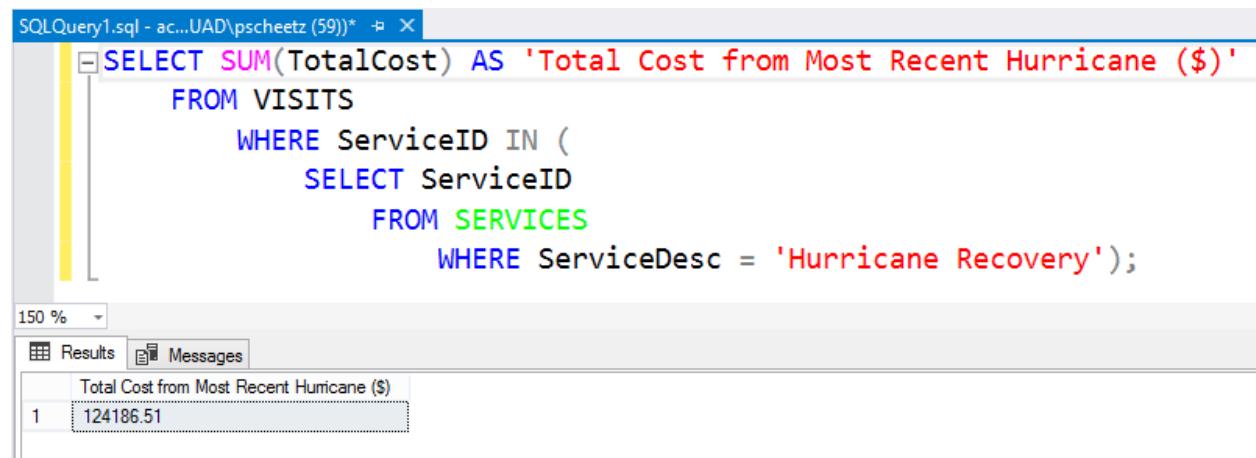
The screenshot shows a SQL Server Management Studio (SSMS) interface. At the top, there is a title bar with the text "SQLQuery1.sql - a...UAD\bmdavi24 (56)*". Below the title bar is a query editor window containing the following SQL code:

```
SELECT SUM(TotalCost) AS 'Total Revenue for 2019'  
FROM VISITS  
WHERE DateOfVisit LIKE '2019%';
```

Below the query editor is a results pane. The top of the results pane has a zoom level of "100 %". It contains two tabs: "Results" and "Messages". The "Results" tab is selected and displays the following output:

| Total Revenue for 2019 | |
|------------------------|------|
| 1 | NULL |

10. Total damages caused by the last hurricane.



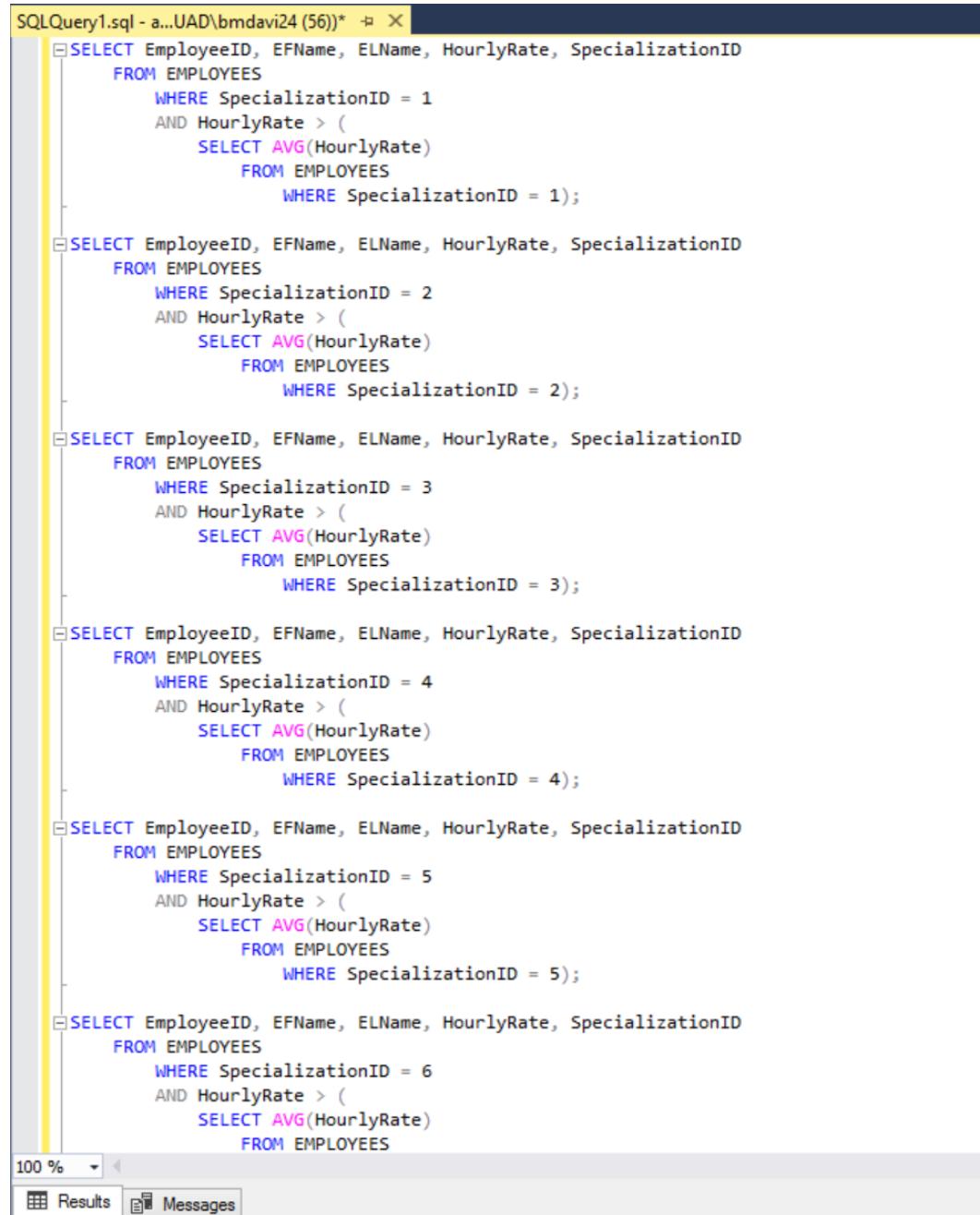
The screenshot shows a SQL Server Management Studio (SSMS) interface. The query window title is "SQLQuery1.sql - ac...UAD\pscheetz (59)*". The query itself is:

```
SELECT SUM(TotalCost) AS 'Total Cost from Most Recent Hurricane ($)'
FROM VISITS
WHERE ServiceID IN (
    SELECT ServiceID
    FROM SERVICES
    WHERE ServiceDesc = 'Hurricane Recovery');
```

The results pane shows one row of data:

| Total Cost from Most Recent Hurricane (\$) |
|--|
| 124186.51 |

11. Employees making more than the average rate for their job specification.



The screenshot shows a SQL query window in SQL Server Management Studio. The title bar reads "SQLQuery1.sql - a...UAD\bmdavi24 (56)*". The main area contains six expandable queries, each selecting employees from the "EMPLOYEES" table based on their specialization ID and hourly rate being greater than the average hourly rate for that specialization. The queries are numbered 1 through 6. The interface includes a vertical scrollbar on the left, a status bar at the bottom showing "100 %", and tabs for "Results" and "Messages" at the bottom right.

```
SELECT EmployeeID, EFName, ELName, HourlyRate, SpecializationID
FROM EMPLOYEES
WHERE SpecializationID = 1
AND HourlyRate > (
    SELECT AVG(HourlyRate)
    FROM EMPLOYEES
    WHERE SpecializationID = 1);

SELECT EmployeeID, EFName, ELName, HourlyRate, SpecializationID
FROM EMPLOYEES
WHERE SpecializationID = 2
AND HourlyRate > (
    SELECT AVG(HourlyRate)
    FROM EMPLOYEES
    WHERE SpecializationID = 2);

SELECT EmployeeID, EFName, ELName, HourlyRate, SpecializationID
FROM EMPLOYEES
WHERE SpecializationID = 3
AND HourlyRate > (
    SELECT AVG(HourlyRate)
    FROM EMPLOYEES
    WHERE SpecializationID = 3);

SELECT EmployeeID, EFName, ELName, HourlyRate, SpecializationID
FROM EMPLOYEES
WHERE SpecializationID = 4
AND HourlyRate > (
    SELECT AVG(HourlyRate)
    FROM EMPLOYEES
    WHERE SpecializationID = 4);

SELECT EmployeeID, EFName, ELName, HourlyRate, SpecializationID
FROM EMPLOYEES
WHERE SpecializationID = 5
AND HourlyRate > (
    SELECT AVG(HourlyRate)
    FROM EMPLOYEES
    WHERE SpecializationID = 5);

SELECT EmployeeID, EFName, ELName, HourlyRate, SpecializationID
FROM EMPLOYEES
WHERE SpecializationID = 6
AND HourlyRate > (
    SELECT AVG(HourlyRate)
    FROM EMPLOYEES
```

We ran the same query for every SpecializationID.

Output from this query:

The screenshot shows a SQL Server Management Studio window with a query results grid. The query is:

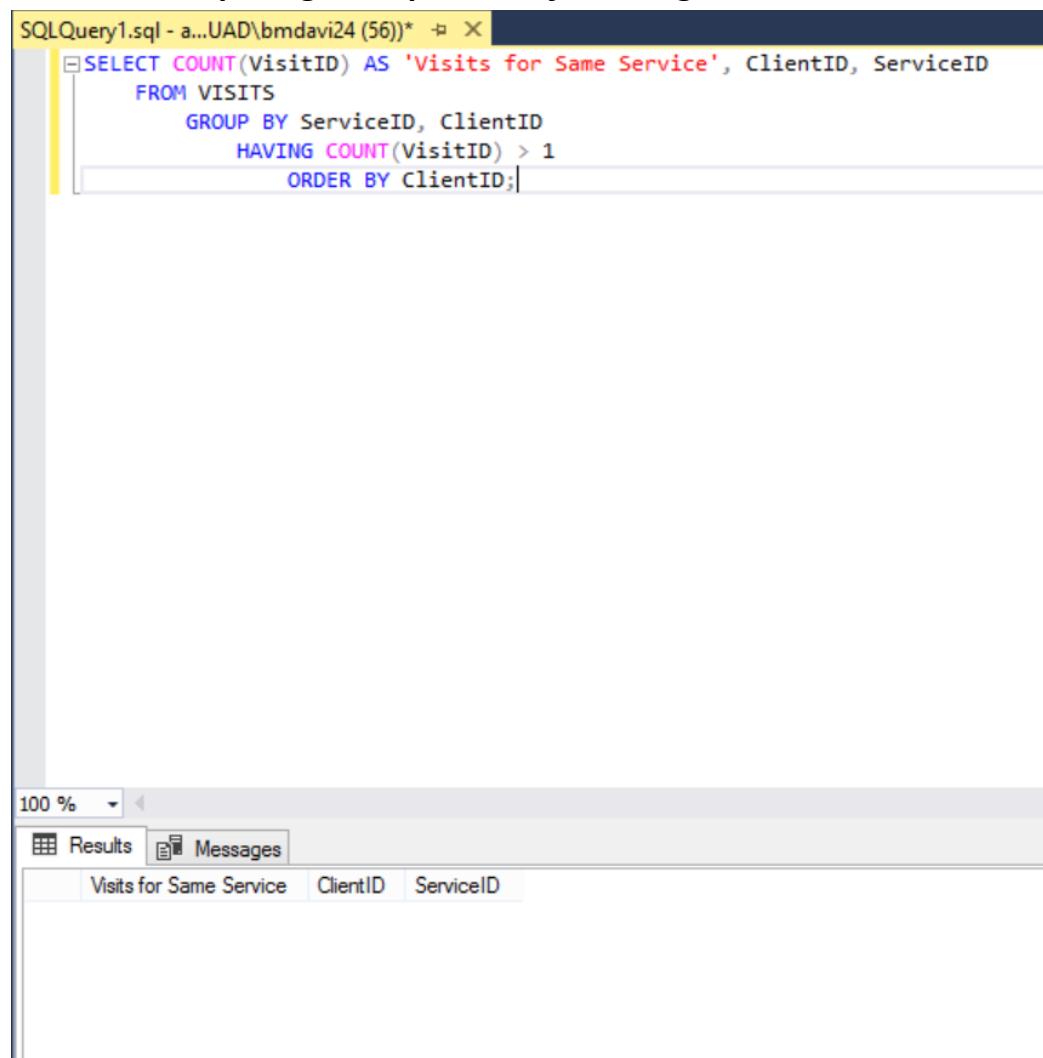
```
SQLQuery1.sql - a...UAD\bmdavi24 (56)* -> X
SELECT EmployeeID, EFName, ELName, HourlyRate, SpecializationID
FROM EMPLOYEES
```

The results grid displays 17 rows of employee data:

| EmployeeID | EFName | ELName | HourlyRate | SpecializationID |
|------------|-------------|------------|-------------|------------------|
| 1 | 977324890 | Sheffy | Hinkley | 73.26 |
| 1 | 229054573 | Clarabelle | Tearle | 82.16 |
| 2 | 251747956 | Aubine | Bemdtsson | 85.97 |
| 1 | 2985053385 | Danica | Kensley | 88.88 |
| 1 | 362079200 | York | Curgeneruer | 88.84 |
| 2 | 4063789551 | Albrecht | Marcum | 71.09 |
| 1 | 4183901477 | Ker | Rebeck | 77.43 |
| 2 | 46340033099 | Emanuel | Comelius | 91.92 |
| 1 | 5477528289 | Durante | Dreier | 76.82 |
| 2 | 572549237 | Ree | Pauler | 54.52 |
| 1 | 5966772740 | Kim | Gagen | 99.60 |
| 1 | 6975164124 | Rob | Rolston | 83.64 |
| 2 | 7583088441 | Gilberta | Way | 91.38 |
| 3 | 7601577954 | Dillon | Jillet | 86.55 |
| 1 | 7895046015 | Dannel | Oyley | 86.13 |
| 2 | 8187560116 | Sheme | McKiddin | 85.79 |
| 1 | 9918294620 | Rollin | Rollins | 81.27 |

At the bottom of the results grid, there is a message bar indicating the query was executed successfully and providing connection information.

12. Clients requiring multiple visits for a single service.



The screenshot shows a SQL Server Management Studio (SSMS) interface. At the top, there is a title bar with the text "SQLQuery1.sql - a...UAD\bmdavi24 (56)*". Below the title bar is a query editor window containing the following T-SQL code:

```
SELECT COUNT(VisitID) AS 'Visits for Same Service', ClientID, ServiceID
FROM VISITS
GROUP BY ServiceID, ClientID
HAVING COUNT(VisitID) > 1
ORDER BY ClientID;
```

Below the query editor is a results pane with a header row containing three columns: "Visits for Same Service", "ClientID", and "ServiceID". The results pane is currently empty, indicating no data has been returned from the query.

Integrity Tests

Entity Integrity Tests

The screenshot shows a SQL Server Management Studio (SSMS) interface. In the top query editor window, a SELECT statement is written to check for null ServiceID values in the SERVICES table:

```
SQLQuery1.sql - a...UAD\bmdavi24 (67)* ➔ X
SELECT *
FROM SERVICES
WHERE ServiceID IS NULL;
```

In the bottom results pane, the tabs "Results" and "Messages" are visible. The "Results" tab is selected, showing a header row with columns: ServiceID, ServiceDesc, and ServiceRate.

SQLQuery1.sql - a...UAD\bmdavi24 (67)* ✎ X

```
SELECT *
FROM EMPLOYEES
WHERE EmployeeID IS NULL;
```

100 % ◀

Results Messages

| EmployeeID | EFName | ELName | HourlyRate | EAddress | EPhone | SpecializationID | State | City |
|------------|--------|--------|------------|----------|--------|------------------|-------|------|
| | | | | | | | | |

SQLQuery1.sql - a...UAD\bmdavi24 (67)* ✎ X

```
SELECT *
FROM PROPERTIES
WHERE PropertyID IS NULL;
```

100 % ◀

Results Messages

| PropertyID | PropName | Address | #Bed | #Bath | SqFt | YrBuilt | RoofType | ExteriorType | ClientID |
|------------|----------|---------|------|-------|------|---------|----------|--------------|----------|
| | | | | | | | | | |

SQLQuery1.sql - a...UAD\bmdavi24 (67)* ↗ X

```
SELECT *
FROM CLIENTS
WHERE ClientID IS NULL;
```

100 %

Results Messages

| ClientID | CIName | CIAddress | CIPhone | YearsServiced | CIEmail | CIState | CI City |
|----------|--------|-----------|---------|---------------|---------|---------|---------|
|----------|--------|-----------|---------|---------------|---------|---------|---------|

SQLQuery1.sql - a...UAD\bmdavi24 (67)* ✎ X

```
SELECT *
FROM VISITS
WHERE VisitID IS NULL;
```

100 %

Results Messages

| VisitID | DateOfVisit | StartDate | EndDate | Owner | TotalCost | PropertyID | ServiceID | ClientID | EmployeeID |
|---------|-------------|-----------|---------|-------|-----------|------------|-----------|----------|------------|
| | | | | | | | | | |

SQLQuery1.sql - a...UAD\bmdavi24 (67)* X

```
SELECT *
FROM APPOINTMENTS
WHERE AppointmentID IS NULL;
```

100 %

Results Messages

| AppointmentID | VisitID | ServiceID | EmployeeID |
|---------------|---------|-----------|------------|
|---------------|---------|-----------|------------|

SQLQuery1.sql - a...UAD\bmdavi24 (67)* X

```
SELECT *
FROM CORPORATE
WHERE ClientID IS NULL;
```

100 % <

Results Messages

| ContactName | ContactPhone | ClientID |
|-------------|--------------|----------|
|-------------|--------------|----------|

SQLQuery1.sql - a...UAD\bmdavi24 (67)* ✎ X

```
SELECT *
FROM INDIVIDUAL
WHERE ClientID IS NULL;
```

100 %

Results Messages

| CreditCard | ClientID |
|------------|----------|
|------------|----------|

Referential Integrity Tests

The screenshot shows a SQL Server Management Studio window with a query editor and a results grid.

Query Editor Content:

```
SQLQuery1.sql - a...UAD\bmdavi24 (67)* SELECT AppointmentID, ServiceDesc, EFName, DateOfVisit  
FROM APPOINTMENTS, SERVICES, EMPLOYEES, VISITS  
WHERE SERVICES.ServiceID = APPOINTMENTS.ServiceID  
AND EMPLOYEES.EmployeeID = APPOINTMENTS.EmployeeID  
AND VISITS.VisitID = APPOINTMENTS.VisitID;
```

Results Grid:

| | AppointmentID | ServiceDesc | EFName | DateOfVisit |
|----|---------------|---|------------|-------------|
| 1 | 5159 | Landscaping | Susi | 2017-04-06 |
| 2 | 17992 | Housekeeping | Leola | 2017-05-09 |
| 3 | 43058 | Standard Inspection (Interior and Exterior) | Sheffy | 2017-01-02 |
| 4 | 48216 | Housekeeping | Durante | 2017-05-09 |
| 5 | 51161 | On-Site Installation | Clarabelle | 2017-05-17 |
| 6 | 60439 | Standard Inspection (Interior) | Gilberta | 2017-04-08 |
| 7 | 62077 | Pest Control | Winifield | 2017-02-14 |
| 8 | 77719 | Pool Service | Larina | 2017-05-18 |
| 9 | 78245 | Standard Inspection (Interior) | York | 2017-03-23 |
| 10 | 112013 | On-Site Installation | Caye | 2017-03-17 |
| 11 | 112384 | Landscaping | Danica | 2017-02-25 |
| 12 | 130592 | Standard Inspection (Exterior) | Ree | 2017-05-15 |
| 13 | 131115 | Standard Inspection (Interior) | York | 2017-01-15 |
| 14 | 140617 | On-Site Delivery Acceptance | Dillon | 2017-04-11 |

Message Bar:

Query executed successfully. | acadsql17.asurite.ad.asu.edu | ASUAD\bmdavi24 (67) | SP20_CIS365_19668_TEAM2 | 00:00:00 | 60 rows

SQLQuery1.sql - a...UAD\bmdavi24 (67)*

```
SELECT VisitID, PropName, CName
FROM VISITS, PROPERTIES, CLIENTS
WHERE VISITS.PropertyID = PROPERTIES.PropertyID
AND VISITS.ClientID = CLIENTS.ClientId;
```

100 %

Results Messages

| | VisitID | PropName | CName |
|----|---------|-----------|----------------------|
| 1 | 8453 | Oneal | Modesto Oneal |
| 2 | 150336 | Williams | Fredricka Williams |
| 3 | 28319 | Macias | Tyrone Macias |
| 4 | 489302 | Klein | Alena Klein |
| 5 | 986315 | Foster | Arianne Foster |
| 6 | 752306 | Cannon | Vemon Cannon |
| 7 | 401071 | Delacruz | Cody Delacruz |
| 8 | 189333 | Dougherty | Kristopher Dougherty |
| 9 | 506920 | Wade | Ahmed Wade |
| 10 | 930329 | Johnson | Elton Johnson |
| 11 | 26217 | Graham | Sparkle Graham |
| 12 | 54238 | Beck | Shirly Beck |
| 13 | 996620 | Hansen | Carlo Hansen |
| 14 | 833698 | Castaneda | Beverlee Castaneda |

Query executed successfully. | acadsql17.asurite.ad.asu.edu | ASUAD\bmdavi24 (67) | SP20_CIS365_19668_TEAM2 | 00:00:00 | 30 rows

SQLQuery1.sql - a...UAD\bmdavi24 (67)* ✎ X

```
SELECT CLIENTS.ClientID, ContactName, CreditCard
FROM CORPORATE, INDIVIDUAL, CLIENTS
WHERE CLIENTS.ClientID = CORPORATE.ClientID
AND INDIVIDUAL.ClientID = CLIENTS.ClientId;
```

100 %

Results Messages

| ClientID | ContactName | CreditCard |
|----------|-------------|------------|
|----------|-------------|------------|

Basic Tests

Basic Test 1

SQLQuery1.sql - ac...AD\bmdavi24 (136)* X

```

/* Basic Test #1 - Adding a new client with no last name (e.g. a corporation) and a Canadian zip code (6 letters and numbers). This query should succeed.
BEFORE SCREENSHOT */

SELECT *
FROM CLIENTS
WHERE CLName IS NULL
AND ClZip LIKE 'a%';

/* Basic Test #1 - Adding a new client with no last name (e.g. a corporation) and a Canadian zip code (6 letters and numbers). This query should succeed.
AFTER SCREENSHOT */

INSERT INTO CLIENTS
VALUES (2738192, '123 South 5th St', 1827762154, 2, 'djskdoe@gmail.com', 'AZ', 'Tempe', 'a12b12', 'Walmart', NULL);

```

100 %

Results Messages

| ClientID | CIAddress | CIPhone | YearsServiced | CEEmail | CIState | CICity | ClZip | CFName | CLName |
|----------|-----------|---------|---------------|---------|---------|--------|-------|--------|--------|
|----------|-----------|---------|---------------|---------|---------|--------|-------|--------|--------|

SQLQuery1.sql - ac...AD\bmdavi24 (136)* X

```

/* Basic Test #1 - Adding a new client with no last name (e.g. a corporation) and a Canadian zip code (6 letters and numbers). This query should succeed.
BEFORE SCREENSHOT */

SELECT *
FROM CLIENTS
WHERE CLName IS NULL
AND ClZip LIKE 'a%';

/* Basic Test #1 - Adding a new client with no last name (e.g. a corporation) and a Canadian zip code (6 letters and numbers). This query should succeed.
AFTER SCREENSHOT */

INSERT INTO CLIENTS
VALUES (2738192, '123 South 5th St', 1827762154, 2, 'djskdoe@gmail.com', 'AZ', 'Tempe', 'a12b12', 'Walmart', NULL);

```

100 %

Results Messages

| ClientID | CIAddress | CIPhone | YearsServiced | CEEmail | CIState | CICity | ClZip | CFName | CLName | |
|----------|-----------|------------------|---------------|---------|-------------------|--------|-------|--------|---------|------|
| 1 | 2738192 | 123 South 5th St | 1827762154 | 2 | djskdoe@gmail.com | AZ | Tempe | a12b12 | Walmart | NULL |

Basic Test 2

```
SQLQuery1.sql - a...UAD\bmdavi24 (56)* # X
/* Basic Test #2 - Adding a new client without a phone number. This query should NOT succeed */

SELECT *
FROM CLIENTS
WHERE ClPhone is NULL; /* Showing that there are no rows with a NULL phone
number to begin with */
```

100 % <

Results Messages

| ClientID | ClAddress | ClPhone | YearsServiced | ClEmail | ClState | ClCity | CFName | CLName | ClZip |
|----------|-----------|---------|---------------|---------|---------|--------|--------|--------|-------|
| | | | | | | | | | |

```
SQLQuery1.sql - a...UAD\bmdavi24 (56)* # X
/* Basic Test #2 - Adding a new client without a phone number. This query should NOT succeed */

SELECT *
FROM CLIENTS
WHERE ClPhone is NULL; /* Showing that there are no rows with a NULL phone
number to begin with */

INSERT INTO CLIENTS
VALUES (128912, '123 S Door Ave', NULL, 20, 'ahhhhhh@gmail.com', 'AZ', 'Tucson', 'Melanie', 'Scott', 82718);
/* showing that we are unable to insert a row with a NULL phone number */
```

100 % <

Messages

```
Msg 515, Level 16, State 2, Line 8
Cannot insert the value NULL into column 'ClPhone', table 'SP20_CIS365_19668_TEAM2.dbo.CLIENTS'; column does not allow nulls. INSERT failed.
The statement has been terminated.
```

Basic Test 3

The screenshot shows the SSMS interface with two tabs open: 'SQLQuery1.sql' and 'SQLQuery2.sql'. The 'Results' tab is selected, displaying the output of the following query:

```
select * from visits
```

The results are presented in a table with the following columns: VisitID, DateOfVisit, StartDate, EndDate, Owner, TotalCost, PropertyID, ServiceID, ClientID, EmployeeID. The data consists of 30 rows, each representing a visit record. The last row, VisitID 681771, is highlighted in blue.

| VisitID | DateOfVisit | StartDate | EndDate | Owner | TotalCost | PropertyID | ServiceID | ClientID | EmployeeID | |
|---------|-------------|------------|------------|------------|-------------------|------------|-----------|----------|------------|------------|
| 16 | 773510 | 2017-05-29 | 2017-10-08 | 2019-09-03 | Maxie Daugherty | 34098.94 | 8448 | 1 | 64593989 | 5804237177 |
| 17 | 937667 | 2017-04-08 | 2017-11-03 | 2019-12-31 | Madelaine Dunn | 97975.29 | 4879 | 1 | 93971647 | 977324890 |
| 18 | 266008 | 2017-05-17 | 2017-04-16 | 2018-09-03 | Laurene Herring | 74936.77 | 4855 | 5 | 65321745 | 229054573 |
| 19 | 799045 | 2017-01-02 | 2017-06-09 | 2018-12-09 | Cary Duarte | 44753.40 | 5726 | 3 | 79147483 | 5864813592 |
| 20 | 388938 | 2017-05-23 | 2017-04-25 | 2019-06-08 | Rodrigo Melton | 44289.93 | 1811 | 4 | 95811088 | 8187560116 |
| 21 | 417991 | 2017-03-23 | 2017-02-12 | 2018-12-18 | Elinor Gay | 37963.65 | 3190 | 2 | 95264136 | 5477528289 |
| 22 | 391378 | 2017-02-14 | 2017-05-20 | 2019-06-17 | Dodie Ford | 97281.34 | 8770 | 7 | 83443310 | 5840844404 |
| 23 | 130047 | 2017-05-09 | 2017-11-16 | 2019-02-11 | Rigoberto Finley | 52230.82 | 8901 | 6 | 18896384 | 4063789551 |
| 24 | 840222 | 2017-02-11 | 2017-01-21 | 2018-09-17 | Carol Vasquez | 86354.72 | 5882 | 2 | 3548957 | 2075868651 |
| 25 | 10384 | 2017-01-15 | 2017-11-05 | 2019-07-18 | Lucius Bowers | 90846.03 | 5754 | 1 | 20536872 | 2627959571 |
| 26 | 241683 | 2017-01-09 | 2017-07-21 | 2018-05-24 | Matthew Mcintyre | 28903.56 | 2128 | 1 | 80920878 | 6975164124 |
| 27 | 501481 | 2017-04-06 | 2017-09-20 | 2018-09-09 | Charisse Braun | 46830.73 | 9051 | 9 | 39862515 | 251747956 |
| 28 | 627588 | 2017-01-26 | 2017-10-07 | 2020-02-25 | Kenyatta Carroll | 40056.08 | 7059 | 7 | 50655329 | 7895046015 |
| 29 | 350883 | 2017-02-02 | 2017-06-17 | 2020-02-08 | Efren Byrd | 48206.24 | 6717 | 3 | 4812308 | 7583088441 |
| 30 | 681771 | 2017-01-16 | 2017-02-14 | 2018-11-20 | Collette Clements | 77142.88 | 9851 | 2 | 87915581 | 6918216884 |

Query executed successfully.

acdsql17.asurite.ad.asu.edu | ASUAD\pscheetz (57) | SP20_CIS365_19668_TEAM2 | 00:00:00 | 30 rows

Ln 30 Col 1 INS

SQLQuery1.sql - ac...UAD\pscheetz (57)* → SQLQuery2.sql - ac...UAD\pscheetz (64)*

```

/*
 * Basic Test #3
 * Deleting a visit and all visit details should succeed using one statement.
 */

begin transaction
    delete from VISITS
    where VisitID = 681771;
    --rollback;

select * from VISITS

```

16 %

| | VisitID | DateOfVisit | StartDate | EndDate | Owner | TotalCost | PropertyID | ServiceID | ClientID | EmployeeID |
|----|---------|-------------|------------|------------|------------------|-----------|------------|-----------|----------|------------|
| 15 | 4761 | 2017-04-27 | 2017-06-09 | 2019-03-03 | Brett Randall | 61540.12 | 4270 | 1 | 66233683 | 3492369222 |
| 16 | 773510 | 2017-05-29 | 2017-10-08 | 2019-09-03 | Maxie Daugherty | 34098.94 | 8448 | 1 | 64693989 | 5804237177 |
| 17 | 937667 | 2017-04-08 | 2017-11-03 | 2019-12-31 | Madelaine Dunn | 97975.29 | 4879 | 1 | 93971647 | 977324890 |
| 18 | 266008 | 2017-05-17 | 2017-04-16 | 2018-09-03 | Laurene Herring | 74936.77 | 4855 | 5 | 65321745 | 229054573 |
| 19 | 799045 | 2017-01-02 | 2017-06-09 | 2018-12-09 | Cary Duarte | 44753.40 | 5726 | 3 | 79147483 | 5864813592 |
| 20 | 388938 | 2017-05-23 | 2017-04-25 | 2019-06-08 | Rodrigo Melton | 44289.93 | 1811 | 4 | 95811088 | 8187560116 |
| 21 | 417991 | 2017-03-23 | 2017-02-12 | 2018-12-18 | Elinor Gay | 37963.65 | 3190 | 2 | 95264136 | 5477528289 |
| 22 | 391378 | 2017-02-14 | 2017-05-20 | 2019-06-17 | Dodie Ford | 97281.34 | 8770 | 7 | 83443310 | 5840844404 |
| 23 | 130047 | 2017-05-09 | 2017-11-16 | 2019-02-11 | Rigoberto Finley | 52230.82 | 8901 | 6 | 18896384 | 4063789551 |
| 24 | 840222 | 2017-02-11 | 2017-01-21 | 2018-09-17 | Carol Vasquez | 86354.72 | 5882 | 2 | 3548957 | 2075868651 |
| 25 | 10384 | 2017-01-15 | 2017-11-05 | 2019-07-18 | Lucius Bowers | 90846.03 | 5754 | 1 | 20536872 | 2627959571 |
| 26 | 241683 | 2017-01-08 | 2017-07-21 | 2018-05-24 | Matthew Mcintyre | 28903.56 | 2128 | 1 | 80920878 | 6975164124 |
| 27 | 501481 | 2017-04-06 | 2017-09-20 | 2018-09-09 | Charisse Braun | 46830.73 | 9051 | 9 | 39862515 | 251747956 |
| 28 | 627588 | 2017-01-26 | 2017-10-07 | 2020-02-25 | Kenyatta Carroll | 40056.08 | 7059 | 7 | 50655329 | 7895046015 |
| 29 | 350883 | 2017-02-02 | 2017-06-17 | 2020-02-08 | Efren Byrd | 48206.24 | 6717 | 3 | 4812308 | 7583088441 |

Query executed successfully.

acadsq117.asurite.ad.asu.edu | ASUAD\pscheetz (57) | SP20_CIS365_19668_TEAM2 | 00:00:00 | 29 rows

Ln 29 Col 1 INS

Basic Test 4

SQLQuery4.sql - a...AD\bmbenne6 (57)*

```
select * from clients;
select * from properties;
```

Results

| ClientID | ClientAddress | ClientPhone | YearsServiced | ClientEmail | ClientState | ClientCity | ClientName | ClientName | ClientZip | |
|----------|--------------------|--------------------------|---------------|------------------|-------------------------|------------|---------------|------------|-----------|-------|
| 10 | 936 Amoth Crossing | 7042993362 | 23 | stur3@tumblr.com | NC | Charlotte | Alena | Hein | 11865 | |
| 11 | 18896384 | 266 Weeping Birch Circle | 3121237328 | 17 | smurthwatem@blklist.com | IL | Chicago | Rigoberto | Finley | 29529 |
| 12 | 20536872 | 441 South Avenue | 7162972323 | 24 | mcantero@loc.gov | NY | Buffalo | Lucius | Bowers | 72504 |
| 13 | 36328797 | 2946 West Lane | 4058676001 | 23 | salesb@wikispaces.com | OK | Oklahoma City | Shirly | Beck | 85409 |
| 14 | 39062515 | 1220 8th Plaza | 9379393341 | 15 | sbohardq@wikispaces.com | OH | Springfield | Charisse | Braun | 85651 |
| 15 | 45730150 | 453 Miller Crossing | 7065809309 | 20 | lmewitt@meetup.com | GA | Cumming | Sparkle | Graham | 12369 |

| PropertyID | PropertyName | Address | #Bed | #Bath | SqFt | YrBuilt | RoofType | ExteriorType | ClientID |
|------------|--------------|-------------------------------|------|-------|----------|---------|----------|--------------|----------|
| 5 | Foster | 53675 Alpine Street | 7 | 7 | 22937.05 | 2005 | shingles | brick | 81164013 |
| 6 | Cannon | 5 Brown Trail | 5 | 10 | 2533.40 | 2008 | shingles | viny | 68603197 |
| 7 | 6434 | Delacruz 3 Myrtle Park | 5 | 9 | 6046.60 | 1995 | tile | brick | 91761429 |
| 8 | 4630 | Dougherty 1 Dexter Crossing | 9 | 3 | 49104.89 | 1998 | tile | viny | 55034572 |
| 9 | 9350 | Wade 09320 Hansons Place | 4 | 10 | 92979.52 | 1999 | shingles | brick | 83696889 |
| 10 | 5555 | Johnson 4838 Lakeland Terrace | 5 | 3 | 76643.08 | 2013 | concrete | stucco | 7101753 |
| 11 | 9166 | Graham 453 Miller Crossings | 2 | 6 | 17194.44 | 1996 | tile | viny | 4K7n14n |

SQLQuery4.sql - a...AD\bmbenne6 (57)*

```
select * from clients;
select * from properties;
```

```
insert into clients values (5000000, '123 Main St', 5555555555, 1, 'test@test.com', 'AZ', 'Phoenix', 'Test', 'Subject', 10000)
insert into properties values (1000, 'Subject', '123 Main St', 1, 1, 1111.11, 2019, 'tile', 'brick', 5000000)
```

Results

| | | | | | | | | | | |
|---|---------|-----------------------|------------|----|--------------------------|----|---------------|---------|---------|-------|
| 4 | 4812308 | 148 Transport Parkway | 2146294515 | 19 | cdunstans@wikipedia.org | TX | Dallas | Efren | Byrd | 69378 |
| 5 | 5000000 | 123 Main St | 5555555555 | 1 | test@test.com | AZ | Phoenix | Test | Subject | 10000 |
| 6 | 6586146 | 8 Km Pass | 2813112153 | 2 | measonu@shareasale.com | TX | Houston | Carolin | Johnson | 40226 |
| 7 | 7101753 | 4838 Lakeland Terrace | 7245078034 | 5 | cskatcher9@behance.net | PA | New Castle | Eton | Johnson | 74842 |
| 8 | 8565248 | 35 Tony Avenue | 2127324383 | 12 | mshelum2@marketwatch.com | NY | New York City | Tyrone | Macias | 50822 |

| PropertyID | PropertyName | Address | #Bed | #Bath | SqFt | YrBuilt | RoofType | ExteriorType | ClientID |
|------------|--------------|------------------------------|------|-------|----------|---------|----------|--------------|----------|
| 26 | 7059 | Carroll 2979 Sundown Terrace | 9 | 10 | 30238.21 | 2005 | shingles | stucco | 50655329 |
| 27 | 6717 | Byrd 148 Transport Parkway | 9 | 8 | 82821.34 | 2001 | concrete | brick | 4812308 |
| 28 | 9851 | Clements 26 Lakeland Way | 9 | 8 | 40206.13 | 1999 | concrete | stucco | 87915581 |
| 29 | 1106 | Johnson 8 Km Pass | 5 | 8 | 49375.71 | 2003 | tile | stucco | 6586146 |
| 30 | 6789 | Andrews 229 David Point | 6 | 1 | 38233.35 | 2007 | shingles | viny | 68203269 |
| 31 | 4270 | Randall 902 Lotheville Road | 8 | 9 | 55927.76 | 1997 | tile | stucco | 66233683 |
| 32 | 8448 | Daugherty 17 Fulton Road | 3 | 7 | 32394.89 | 2009 | shingles | stucco | 64593989 |
| 33 | 1000 | Subject 123 Main St | 1 | 1 | 1111.11 | 2019 | tile | brick | 5000000 |

Basic Test 5

```
SQLQuery4.sql - a...AD\bmbenne6 (57)* - X
select * from clients;
select * from properties;
```

100 % ▾

Results Messages

| | ClientID | CIAddress | CIPhone | YearsServiced | CEmail | COState | COCity | CFName | CLName | COZip |
|----|------------|--------------------------|-----------------------|---------------|----------------------------|----------|---------------|-----------|--------------|----------|
| 10 | 9038853 | 936 Amoth Crossing | 7042993362 | 23 | sstur3@tumblr.com | NC | Charlotte | Alena | Klein | 11865 |
| 11 | 18896384 | 266 Weeping Birch Circle | 3121237328 | 17 | smurthwaterm@blinklist.com | IL | Chicago | Rigoberto | Finley | 29529 |
| 12 | 20536872 | 441 South Avenue | 7162972323 | 24 | mcantero@loc.gov | NY | Buffalo | Lucius | Bowers | 72504 |
| 13 | 36328797 | 2946 West Lane | 4058676001 | 23 | salesb@wikispaces.com | OK | Oklahoma City | Shirly | Beck | 85409 |
| 14 | 39862515 | 1220 8th Plaza | 9379333341 | 15 | sboshard@wikispaces.com | OH | Springfield | Charisse | Braun | 85651 |
| 15 | 45730150 | 453 Miller Crossing | 7065809309 | 20 | lmewitt@meetup.com | GA | Cumming | Sparkle | Graham | 12369 |
| | PropertyID | PropName | Address | #Bed | #Bath | SqFt | YrBuilt | RoofType | ExteriorType | ClientID |
| 5 | 7120 | Foster | 53675 Alpine Street | 7 | 7 | 22937.05 | 2005 | shingles | brick | 81164013 |
| 6 | 4133 | Cannon | 5 Brown Trail | 5 | 10 | 2533.40 | 2008 | shingles | vinyl | 68603197 |
| 7 | 6434 | Delacruz | 3 Myrtle Park | 5 | 9 | 6046.60 | 1995 | tile | brick | 91761429 |
| 8 | 4630 | Dougherty | 1 Dexter Crossing | 9 | 3 | 49104.89 | 1998 | tile | vinyl | 55034572 |
| 9 | 9350 | Wade | 09320 Hansons Place | 4 | 10 | 92979.52 | 1999 | shingles | brick | 83696889 |
| 10 | 5555 | Johnson | 4838 Lakeland Terrace | 5 | 3 | 76643.08 | 2013 | concrete | stucco | 7101753 |
| 11 | 9754 | Graham | 4K3 Miller Crossing | 2 | 6 | 17744.44 | 1996 | tile | vinyl | 45730150 |

```
SQLQuery5.sql - a...AD\bmbenne6 (89)* - X SQLQuery4.sql - a...AD\bmbenne6 (57)*
/* Q5. Two similar INSERT statements (#4) executed together in the opposite order should fail. */

insert into properties values (1234, 'Fail', '111 West Ave', 1, 2, 12345.43, 1929, 'concrete', 'vinyl', 1234567)
insert into clients values (1234567, '111 West Ave', 1234567890, 19, 'pleasefail@test.com', 'TX', 'Austin', 'Please', 'Fail', 12345)

select * from properties;
select * from clients;
```

100 % ▾

Messages

```
Msg 547, Level 16, State 0, Line 4
The INSERT statement conflicted with the FOREIGN KEY constraint "FK__PROPERTIES_Clients_SDCAEF64". The conflict occurred in database "SP20_C".
The statement has been terminated.

(1 row(s) affected)
```

Basic Test 6

The screenshot shows a SQL Server Management Studio (SSMS) interface. The top window is titled "SQLQuery2.sql - ac...UAD\pscheetz (59)*" and contains the following SQL script:

```
/* Basic Test #6
   Adding a new employee to complete an existing service should succeed. */

begin transaction
update APPOINTMENTS
    set EmployeeID = 5804237177
        -- set to employee ID 5804237177
        -- was initially 2627959571
    where AppointmentID = 5159

select * from APPOINTMENTS
    where AppointmentID = 5159;

--rollback;
```

The bottom window is titled "Results" and displays the following table:

| | AppointmentID | VisitID | ServiceID | EmployeeID |
|---|---------------|---------|-----------|------------|
| 1 | 5159 | 501481 | 9 | 5804237177 |

Basic Test 7

The screenshot shows two SQL queries in separate windows:

SQLQuery2.sql - a...UAD\acampb20 (84)*

```
>Select * from clients
-- insert fails as null is passed as name
/*
insert into clients
values (34234324, null, '15 Walton Pass', '8608999058', 16, 'hrenton0@ted.com', 'CT', 'Hartford')
*/
```

SQLQuery1.sql - a...UAD\acampb20 (69)*

```
--Select * from clients where clientId = 34234324
-- insert fails as null is passed as name
insert into clients
values (34234324, null, '15 Walton Pass', '8608999058', 16, 'hrenton0@ted.com', 'CT', 'Hartford')
```

Results tab (Top Window):

| ClientID | CIName | CIAddress | CIPhone | YearsServiced | CEmail | CIState | CICity | |
|----------|---------|--------------------|-----------------------|---------------|--------|-------------------------------|--------|-----------|
| 1 | 3496507 | Carlo Hansen | 4 Loomis Point | 4159574029 | 17 | vjakubowicz@howstuffworks.com | CA | San Franc |
| 2 | 3548957 | Carol Vasquez | 9378 Sugar Way | 6024770379 | 16 | zspanswickn@twitpic.com | AZ | Phoenix |
| 3 | 4812308 | Efren Byrd | 148 Transport Parkway | 2146294515 | 19 | cdunstans@wikipedia.org | TX | Dallas |
| 4 | 6586146 | Carolin Johnson | 8 Kim Pass | 2813112153 | 2 | measonu@shareasale.com | TX | Houston |
| 5 | 7101753 | Elton Johnson | 4838 Lakeland Terrace | 7245078034 | 5 | cskatcher9@behance.net | PA | New Cast |
| 6 | 8565248 | Tyrone Macias | 35 Tony Avenue | 2127324383 | 12 | mshellum2@marketwatch.com | NY | New York |
| 7 | 8977331 | Beverlee Castaneda | 17912 Farmco Parkway | 5097900342 | 23 | hswabyd@cdc.gov | WA | Spokane |

Messages tab (Bottom Window):

```
Msg 515, Level 16, State 2, Line 5
Cannot insert the value NULL into column 'CIName', table 'SP20_CIS365_19668_acampb20.dbo.CLIENTS'; column does not allow nulls. INSERT fails.
The statement has been terminated.
```

Basic Test 8

The screenshot shows two SQL queries in the SSMS interface. The top query inserts a new employee record into the EMPLOYEES table:

```
SQLQuery2.sql - a...UAD\acampb20 (84)* ✘ SQLQuery1.sql - a...UAD\acampb20 (69)*
Select * from EMPLOYEES where EmployeeId = 4063789551
/*
INSERT INTO EMPLOYEES
    (EmployeeId, EFName, ELName, HourlyRate, EAddress, EPhone, SpecializationID, State, City)
Values (4063789551,'Sasha','ABC', 50.01, '123 Main S', '1234667989', 1, 'NY','New York City' )

*/
```

The results show one row inserted:

| | EmployeeID | EFName | ELName | HourlyRate | EAddress | EPhone | SpecializationID | State | City |
|---|------------|----------|--------|------------|--------------------|------------|------------------|-------|--------------|
| 1 | 4063789551 | Albrecht | Marcum | 71.09 | 971 Oakridge Drive | 3109325900 | 10 | CA | Santa Monica |

The bottom query attempts to insert the same employee record again, which fails due to a primary key constraint violation:

```
SQLQuery2.sql - a...UAD\acampb20 (84)* ✘ SQLQuery1.sql - a...UAD\acampb20 (69)*
--Select * from EMPLOYEES where EmployeeId = 4063789551
/*
INSERT INTO EMPLOYEES
    (EmployeeId, EFName, ELName, HourlyRate, EAddress, EPhone, SpecializationID, State, City)
Values (4063789551,'Sasha','ABC', 50.01, '123 Main St', '1234667989', 11, 'NY','New York City' )
```

The message pane displays the error:

Msg 2627, Level 14, State 1, Line 3
Violation of PRIMARY KEY constraint 'PK_EMPLOYEE_7AD04FF10F178C0B'. Cannot insert duplicate key in object 'dbo.EMPLOYEES'. The duplicate key value is
The statement has been terminated.

Basic Test 9

SQLQuery2.sql - a...UAD\acampb20 (84)* ➔ X SQLQuery1.sql - a...UAD\acampb20 (69)*

```
Object Explorer
Select distinct SpecializationID from Employees
/*
INSERT INTO EMPLOYEES
    (EmployeeId, EFName, ElName, HourlyRate, EAddress, EPhone, SpecializationID, State, City)
Values (40637895667, 'AnyName' , 'ABC' , 50, '123 Main St', '1234667989', 231231, 'NY','New York City' )
*/
100 %
```

Results Messages

| SpecializationID |
|------------------|
| 1 10 |
| 2 13 |
| 3 14 |
| 4 15 |
| 5 16 |
| 6 17 |
| 7 18 |
| 8 19 |
| 9 21 |
| 10 22 |
| 11 25 |
| 12 26 |
| 13 27 |
| 14 28 |
| 15 29 |
| 16 30 |

SQLQuery3.sql - a...UAD\acampb20 (54)* ➔ X SQLQuery2.sql - a...UAD\acampb20 (84)* SQLQuery1.sql - a...UAD\acampb20 (69)*

```
--Select distinct SpecializationID from Employees where EmployeeId = 40637895667

INSERT INTO EMPLOYEES
    (EmployeeId, EFName, ElName, HourlyRate, EAddress, EPhone, SpecializationID, State, City)
Values (40637895667, 'AnyName' , 'ABC' , 50, '123 Main St', '1234667989', 231231, 'NY','New York City' )
```

100 %

Messages

```
Msg 8115, Level 16, State 5, Line 4
Arithmetic overflow error converting numeric to data type varchar.
The statement has been terminated.
```

Basic Test 10

SQLQuery4.sql - a...UAD\acampb20 (66)* ✘ X SQLQuery3.sql - a...UAD\acampb20 (54)*

```
Select ServiceRate from Services  
-- Insert into Services Values(1,'Cleaing', 1123123222.99)
```

100 %

Results Messages

| | ServiceRate |
|---|-------------|
| 1 | 350.00 |
| 2 | 350.00 |
| 3 | 600.00 |
| 4 | 75.00 |
| 5 | 150.00 |
| 6 | 150.00 |
| 7 | 60.00 |
| 8 | 90.00 |

Query executed successfully. | acadsql17.asurite.ad.asu.edu | ASUAD\acampb20 (66) | SP20_CIS365_19668

SQLQuery4.sql - a...UAD\acampb20 (66)* ✘ X SQLQuery3.sql - a...UAD\acampb20 (54)*

```
--Select ServiceRate from Services  
Insert into Services Values(1,'Cleaing', 1123123222.99)
```

100 %

Messages

Msg 8115, Level 16, State 8, Line 3
Arithmetric overflow error converting numeric to data type numeric.
The statement has been terminated.

100 %

Query completed with errors. | acadsql17.asurite.ad.asu.edu | ASUAD\acampb20 (66) | SP20_CIS365_19668_acam...

Advanced Tests

of Clients per State

SQLQuery2.sql - ac..UAD\pscheetz (59)* X

```
/* Advanced Query #1:  
Find the number of clients per state for every state with more than 1 client.  
Also lists the average years serviced per state.  
*/  
  
select  
    ClState as 'State',  
    count(ClState) as 'Number of Clients',  
    avg(YearsServiced) as 'Average Years Serviced'  
from CLIENTS  
group by ClState  
having count(ClState) > 1; --states with more than 1 client
```

150 % ▶

| | State | Number of Clients | Average Years Serviced |
|---|-------|-------------------|------------------------|
| 1 | AZ | 4 | 8 |
| 2 | CA | 6 | 14 |
| 3 | CT | 3 | 15 |
| 4 | NC | 2 | 16 |
| 5 | NY | 4 | 15 |
| 6 | OH | 2 | 18 |
| 7 | OK | 2 | 18 |
| 8 | PA | 2 | 6 |
| 9 | TX | 4 | 11 |

Number of Services Performed when Performed 5+ Times

SQLQuery2.sql - ac...UAD\pscheetz (59)* ➔ X

```
/* Advanced Query #2:  
Lists the number of services performed by service type,  
where there have been 5 or more services performed  
*/  
  
select  
    S.ServiceDesc as 'Service Type',  
    count(A.ServiceID) as '# of Services Performed'  
from APPOINTMENTS A  
    join SERVICES S  
        on A.ServiceID = S.ServiceID  
group by S.ServiceDesc  
having count(A.ServiceID) >= 5;
```

150 %

| | Service Type | # of Services Performed |
|---|---|-------------------------|
| 1 | Housekeeping | 5 |
| 2 | On-Site Installation | 5 |
| 3 | Pest Control | 5 |
| 4 | Standard Inspection (Exterior) | 11 |
| 5 | Standard Inspection (Interior and Exterior) | 7 |
| 6 | Standard Inspection (Interior) | 19 |

Combined Salaries for Each Specialization

```
SQLQuery2.sql - ac..UAD\pscheetz (59)*  ✎ X
  /* Advanced Query #3
   Lists the total combined salaries for each specialization,
   broken down by state, where the total combined salary is less than $75.
  */

  select
    sum(HourlyRate) as 'Total Hourly Salary',
    SpecializationID as 'Specialization',
    State
  from EMPLOYEES
  group by SpecializationID, State
  having sum(HourlyRate) < 75;
```

150 % ⏪

Results Messages

| | Total Hourly Salary | Specialization | State |
|----|---------------------|----------------|-------|
| 1 | 59.62 | 8 | AZ |
| 2 | 20.00 | 10 | AZ |
| 3 | 32.32 | 1 | CA |
| 4 | 71.09 | 4 | CA |
| 5 | 33.58 | 6 | CA |
| 6 | 57.68 | 5 | CO |
| 7 | 47.14 | 4 | IA |
| 8 | 38.68 | 8 | IL |
| 9 | 54.52 | 6 | NH |
| 10 | 58.03 | 9 | NY |
| 11 | 46.69 | 7 | OH |
| 12 | 10.00 | 1 | TX |
| 13 | 24.48 | 5 | TX |
| 14 | 41.26 | 6 | TX |
| 15 | 73.26 | 1 | WA |

Fully Nested Query 1

The screenshot shows a SQL query window in SQL Server Management Studio. The query is a fully nested SELECT statement:

```
/* Advanced Test #4 - Fully NESTED query (no joins) that combines 3 or more data tables */  
SELECT EPhone  
FROM EMPLOYEES  
WHERE EmployeeID IN (  
    SELECT EmployeeID  
    FROM APPOINTMENTS  
    WHERE ServiceID IN (  
        SELECT ServiceID  
        FROM SERVICES  
        WHERE ServiceRate = 60.00));
```

The results pane displays the output of the query, showing five rows of EPhone numbers:

| EPhone |
|------------|
| 5152085062 |
| 3035498712 |
| 5611606254 |
| 6264824160 |
| 7132344169 |

At the bottom of the results pane, a status bar indicates: "Query executed successfully." and "5 rows".

Fully Nested Query 2

The screenshot shows a SQL query window in SQL Server Management Studio. The query is a fully nested SELECT statement:

```
SQLQuery1.sql - ac...AD\bmdavi24 (136)*  □ X
/* Advanced Test #5 - Fully NESTED query (no joins) that combines 3 or more data tables */
SELECT PropName
  FROM PROPERTIES
 WHERE PropertyID IN (
   SELECT PropertyID
     FROM VISITS
    WHERE ClientID IN (
      SELECT ClientID
        FROM CLIENTS
       WHERE ClPhone = 7042993362));
```

The results pane shows one row of data:

| PropName |
|----------|
| Klein |

At the bottom of the interface, a status bar displays: "Query executed successfully." and "1 rows".

Fully Nested Query 3

The screenshot shows a SQL query window in SQL Server Management Studio. The query is a fully nested SELECT statement:

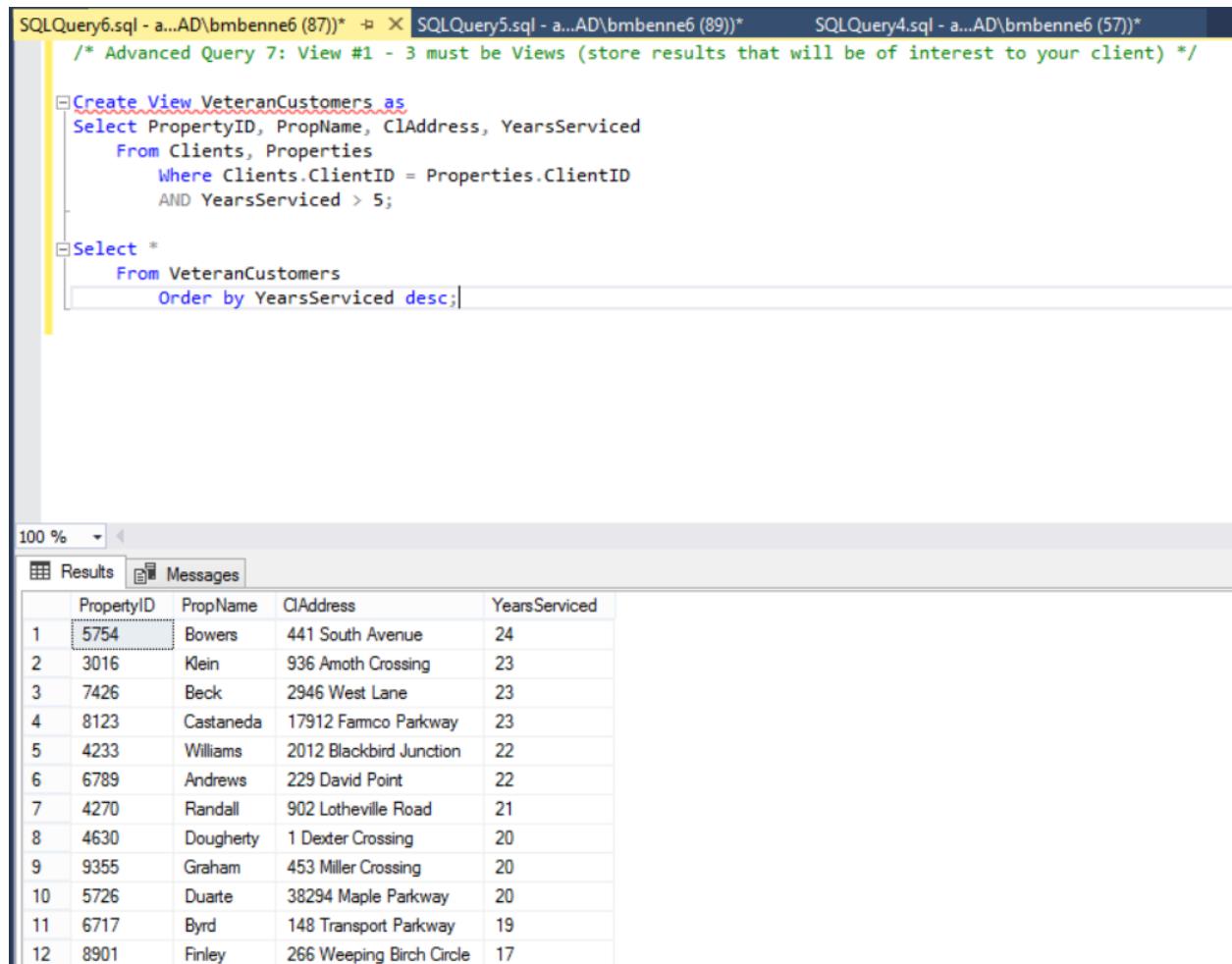
```
/* Advanced Test #6 - Fully NESTED query (no joins) that combines 3 or more data tables */  
SELECT *  
FROM SERVICES  
WHERE ServiceID IN (  
    SELECT ServiceID  
    FROM APPOINTMENTS  
    WHERE VisitID IN (  
        SELECT VisitID  
        FROM VISITS  
        WHERE ClientID IN (  
            SELECT ClientID  
            FROM INDIVIDUAL  
            WHERE CreditCard = 466604156551278)))
```

The results pane shows one row of data:

| | ServiceID | ServiceDesc | ServiceRate |
|---|-----------|-----------------------------|-------------|
| 1 | 4 | On-Site Delivery Acceptance | 75.00 |

At the bottom of the window, a status bar indicates: "Query executed successfully." and "1 rows".

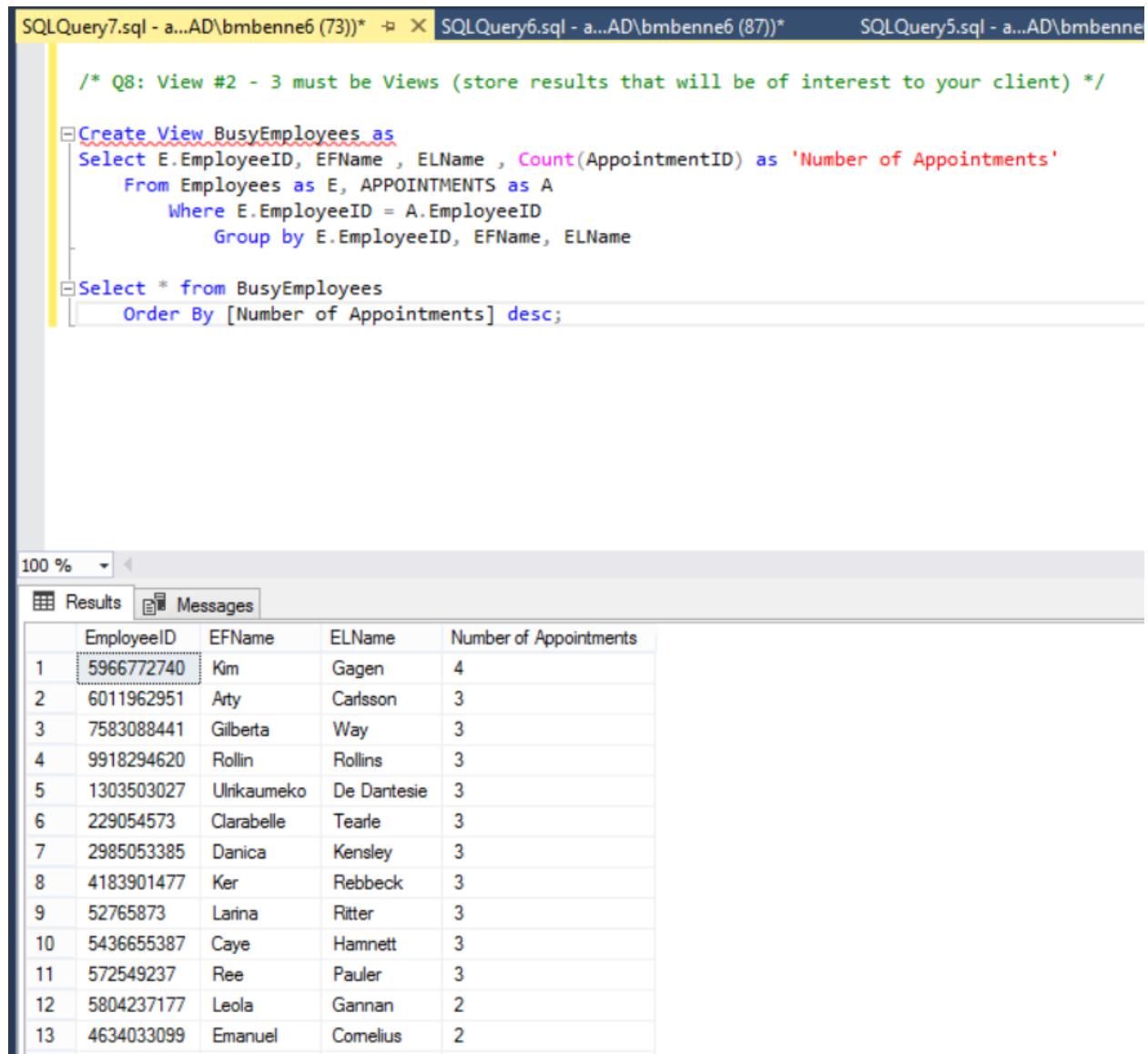
View 1



The screenshot shows a SQL Server Management Studio window. The top tab bar has three tabs: 'SQLQuery6.sql - a...AD\bmbenne6 (87)*' (selected), 'SQLQuery5.sql - a...AD\bmbenne6 (89)*', and 'SQLQuery4.sql - a...AD\bmbenne6 (57)*'. The main pane displays a T-SQL script for creating a view named 'VeteranCustomers'. The script includes a 'Create View' statement and a 'Select' statement with an 'Order by' clause. Below the script, the 'Results' tab is selected, showing a grid of 12 rows of data. The columns are labeled 'PropertyID', 'PropName', 'C1Address', and 'YearsServiced'. The data is as follows:

| | PropertyID | PropName | C1Address | YearsServiced |
|----|------------|-----------|--------------------------|---------------|
| 1 | 5754 | Bowers | 441 South Avenue | 24 |
| 2 | 3016 | Klein | 936 Anoth Crossing | 23 |
| 3 | 7426 | Beck | 2946 West Lane | 23 |
| 4 | 8123 | Castaneda | 17912 Farmco Parkway | 23 |
| 5 | 4233 | Williams | 2012 Blackbird Junction | 22 |
| 6 | 6789 | Andrews | 229 David Point | 22 |
| 7 | 4270 | Randall | 902 Lotheville Road | 21 |
| 8 | 4630 | Dougherty | 1 Dexter Crossing | 20 |
| 9 | 9355 | Graham | 453 Miller Crossing | 20 |
| 10 | 5726 | Duarte | 38294 Maple Parkway | 20 |
| 11 | 6717 | Byrd | 148 Transport Parkway | 19 |
| 12 | 8901 | Finley | 266 Weeping Birch Circle | 17 |

View 2



The screenshot shows a SQL Server Management Studio window with three tabs at the top: 'SQLQuery7.sql - a...AD\bmbenne6 (73)*' (highlighted in yellow), 'SQLQuery6.sql - a...AD\bmbenne6 (87)*', and 'SQLQuery5.sql - a...AD\bmbenne'. The main area contains a SQL query for creating a view named 'BusyEmployees'.

```
/* Q8: View #2 - 3 must be Views (store results that will be of interest to your client) */

Create View BusyEmployees as
Select E.EmployeeID, EFName , ELName , Count(AppointmentID) as 'Number of Appointments'
  From Employees as E, APPOINTMENTS as A
    Where E.EmployeeID = A.EmployeeID
      Group by E.EmployeeID, EFName, ELName

Select * from BusyEmployees
  Order By [Number of Appointments] desc;
```

The results grid below displays 13 rows of data:

| | EmployeeID | EFName | ELName | Number of Appointments |
|----|------------|-------------|-------------|------------------------|
| 1 | 5966772740 | Kim | Gagen | 4 |
| 2 | 6011962951 | Arty | Carlsson | 3 |
| 3 | 7583088441 | Gilberta | Way | 3 |
| 4 | 9918294620 | Rollin | Rollins | 3 |
| 5 | 1303503027 | Ulrikaumeko | De Dantesie | 3 |
| 6 | 229054573 | Clarabelle | Tearle | 3 |
| 7 | 2985053385 | Danica | Kensley | 3 |
| 8 | 4183901477 | Ker | Rebeck | 3 |
| 9 | 52765873 | Larina | Ritter | 3 |
| 10 | 5436655387 | Caye | Hamnett | 3 |
| 11 | 572549237 | Ree | Pauler | 3 |
| 12 | 5804237177 | Leola | Gannan | 2 |
| 13 | 4634033099 | Emanuel | Comelius | 2 |

View 3

```
SQLQuery9.sql - a...AD\bmbenne6 (111)* SQLQuery8.sql - a...AD\bmbenne6 (60)* × SQLQuery5.sql - a...AD\bmbenne6 (89)* SQLQuery4.s
/* Q9. View #3 - 3 must be Views (store results that will be of interest to your client) */
/* Finding the total cost of modern homes (Built in 2000 or newer) that are in the same state as Rodrigo Melton */

Create View ModernHomeServiceCost as
Select Owner , YrBuilt as 'Year Built' , ClCity as 'City' , ClState as 'State' , TotalCost
From VISITS, PROPERTIES, CLIENTS
Where CLIENTS.ClientID = VISITS.ClientID
AND PROPERTIES.PropertyID = VISITS.PropertyID
AND ClState in (Select ClState from CLIENTS where CFName = 'Rodrigo' AND CLName = 'Melton')
AND YrBuilt > 1999;

Select * From ModernHomeServiceCost
Order by TotalCost desc;
```

100 %

| | Owner | Year Built | City | State | TotalCost |
|---|--------------------|------------|---------------|-------|-----------|
| 1 | Fredricka Williams | 2006 | Bakersfield | CA | 90087.57 |
| 2 | Carlo Hansen | 2010 | San Francisco | CA | 27828.25 |
| 3 | Arianne Foster | 2005 | Salinas | CA | 26578.87 |

Transactions

Transaction 1

```
SQLQuery3.sql - a...UAD\acampb20 (91)*      SQLQuery2.sql - a...UAD\acampb20 (89))      SQLQueryTransacti...AD\acampb20 (86)* ↵ X

declare @clientId INT,@propertyId INT

SELECT @clientId = MAX(ClientID) + 1 FROM CLIENTS;
SELECT @propertyId = MAX([PropertyID]) + 1 FROM PROPERTIES;

SELECT * FROM CLIENTS WHERE CLIENTID=@clientId
SELECT * FROM PROPERTIES WHERE PropertyID=@propertyId

BEGIN TRANSACTION _Test

INSERT INTO CLIENTS([ClientID],[CIName],[CIAddress],[CIPhone],[YearsServiced],[CIEmail],[CIState],[CICity])
VALUES(@clientId, 'Testing', '12 George Street', '1234567891',20,'testing@yahoo.com','AZ','PHOENIX')
INSERT INTO PROPERTIES([PropertyID],[PropName],[Address],[#Bed],[#Bath],[SqFt],[YrBuilt],[RoofType],[ExteriorType],[ClientID])
VALUES(@propertyId, 'xyz', '12 Square Road', 5, 3, 3456.3, null, null, null, null, @clientId)
    column #Bed(int, not null)

ROLLBACK TRANSACTION _Test
--COMMIT TRANSACTION _Test

SELECT * FROM CLIENTS WHERE CLIENTID=@clientId
SELECT * FROM PROPERTIES WHERE PropertyID=@propertyId
```

100 %

Results Messages

| ClientID | CIName | CIAddress | CIPhone | YearsServiced | CIEmail | CIState | CICity |
|----------|--------|-----------|---------|---------------|---------|---------|--------|
|----------|--------|-----------|---------|---------------|---------|---------|--------|

| PropertyID | PropName | Address | #Bed | #Bath | SqFt | YrBuilt | RoofType | ExteriorType | ClientID |
|------------|----------|---------|------|-------|------|---------|----------|--------------|----------|
|------------|----------|---------|------|-------|------|---------|----------|--------------|----------|

| ClientID | CIName | CIAddress | CIPhone | YearsServiced | CIEmail | CIState | CICity |
|----------|--------|-----------|---------|---------------|---------|---------|--------|
|----------|--------|-----------|---------|---------------|---------|---------|--------|

| PropertyID | PropName | Address | #Bed | #Bath | SqFt | YrBuilt | RoofType | ExteriorType | ClientID |
|------------|----------|---------|------|-------|------|---------|----------|--------------|----------|
|------------|----------|---------|------|-------|------|---------|----------|--------------|----------|

Query executed successfully. | acadsql17.asurite.ad.asu.edu...

SQLQuery3.sql - a...UAD\acampb20 (91)* SQLQuery2.sql - a...UAD\acampb20 (89)) SQLQueryTransacti...AD\acampb20 (86)*

```

declare @clientId INT,@propertyId INT

SELECT @clientId = MAX(ClientID) + 1 FROM CLIENTS;
SELECT @propertyId = MAX([PropertyID]) + 1 FROM PROPERTIES;

SELECT * FROM CLIENTS WHERE CLIENTID=@clientId
SELECT * FROM PROPERTIES WHERE PropertyID=@propertyId

BEGIN TRANSACTION _Test

INSERT INTO CLIENTS([ClientID],[CIName],[CIAddress],[CIPhone],[YearsServiced],[CIEmail],[CIState],[CICity])
VALUES(@clientId, 'Testing', '12 George Street', '1234567891', 20, 'testing@yahoo.com', 'AZ', 'PHOENIX')
INSERT INTO PROPERTIES([PropertyID],[PropName],[Address],[#Bed],[#Bath],[SqFt],[YrBuilt],[RoofType],[ExteriorType],[ClientID])
VALUES(@propertyId, 'xyz', '12 Square Road', 5, 3, 3456.35, 2008, 'tile', 'brick',@clientId)

--ROLLBACK TRANSACTION _Test
COMMIT TRANSACTION _Test

SELECT * FROM CLIENTS WHERE CLIENTID=@clientId
SELECT * FROM PROPERTIES WHERE PropertyID=@propertyId

```

100 %

Results Messages

| ClientID | CIName | CIAddress | CIPhone | YearsServiced | CIEmail | CIState | CICity | |
|----------|----------|-----------|------------------|---------------|---------|-------------------|--------|---------|
| 1 | 95811091 | Testing | 12 George Street | 1234567891 | 20 | testing@yahoo.com | AZ | PHOENIX |

| PropertyID | PropName | Address | #Bed | #Bath | SqFt | YrBuilt | RoofType | ExteriorType | ClientID | |
|------------|----------|---------|----------------|-------|------|---------|----------|--------------|----------|----------|
| 1 | 9854 | xyz | 12 Square Road | 5 | 3 | 3456.35 | 2008 | tile | brick | 95811091 |

Query executed successfully. | acadsql17.asurite.ad.asu.edu...

Transaction 2

```
SQLQuery6.sql - a...AD\bmbenne6 (103)* X SQLQuery5.sql - a...AD\bmbenne6 (87)*      SQLQuery3.sql - a...AD\bmbenne6 (85)*      SQLQuery2.sql - a...AD\bmbenne6 (84)*
1  /* 2. (5) When adding a visit request, also include what is requested in the service request table. */
2
3 BEGIN TRANSACTION;
4
5 Insert Into VISITS Values (999999, '2017-06-05', '2017-07-02', '2017-08-24', 'Taylor Swift', 100.00, 1001, 1, 1000000, 1000000000);
6 Insert Into APPOINTMENTS Values (424242, 681771, 10, 4063789551);
7
8 Select * From VISITS Where VisitID = 999999
9 Select * From APPOINTMENTS Where AppointmentID = 424242;
10
11 Rollback;
12 Commit;
```

100 %

Results Messages

| | VisitID | DateOfVisit | StartDate | EndDate | Owner | TotalCost | PropertyID | ServiceID | ClientID | EmployeeID |
|---|---------|-------------|------------|------------|--------------|-----------|------------|-----------|----------|------------|
| 1 | 999999 | 2017-06-05 | 2017-07-02 | 2017-08-24 | Taylor Swift | 100.00 | 1001 | 1 | 1000000 | 1000000000 |

| | AppointmentID | VisitID | ServiceID | EmployeeID |
|---|---------------|---------|-----------|------------|
| 1 | 424242 | 681771 | 10 | 4063789551 |

Trigger

```
SQLQuery4.sql - ac...UAD\pscheetz (83)* + X SQLQuery3.sql - ac...AD\pscheetz (136)* SQLQuery1.sql - ac...UAD\pscheetz (74)*
/* 1. (10) TRIGGER: Process a service (visit) request. When a new request is added,
the trigger will automatically assign an employee who has the required specialization
(if the work is plumbing, assign the corresponding employee with a plumber specialization). */

create trigger Appointment_Specialization
on APPOINTMENTS
after insert
as
update APPOINTMENTS
set EmployeeID =
((select EmployeeID
from EMPLOYEES
where SpecializationID in
(select ServiceID
from INSERTED)))
from APPOINTMENTS
join INSERTED I on I.EmployeeID = APPOINTMENTS.EmployeeID
where APPOINTMENTS.AppointmentID = I.AppointmentID;

begin transaction
insert into APPOINTMENTS values (42424242, 681771, 10, 1303503027);

rollback; --execute if data is incorrect
commit; --only execute if data is correct and ready to be implemented

select * from APPOINTMENTS where AppointmentID = 42424242;
select * from employees where SpecializationID = 10
```

135 %

Results Messages

| | AppointmentID | VisitID | ServiceID | EmployeeID |
|---|---------------|---------|-----------|------------|
| 1 | 42424242 | 681771 | 10 | 4063789551 |

| | EmployeeID | EFName | ELName | HourlyRate | EAddress | EPHONE | SpecializationID | State | City |
|---|------------|----------|--------|------------|--------------------|------------|------------------|-------|--------------|
| 1 | 4063789551 | Albrecht | Marcum | 71.09 | 971 Oakridge Drive | 3109325900 | 10 | CA | Santa Monica |

Query executed successfully.

acadsq17.asurite.ad.asu.edu | ASUAD\pscheetz (83) | SP20_CIS365_19668_TEAM2 | 00:00:00 | 2 rows

Procedure

Code part 1 (doesn't fit on a single page)

```
SQLQuery2.sql - ac...UAD\pscheetz (59)*  X
/*
  2. (10) PROCEDURE: When an employee has been assigned to perform service and completes that
  service, add a new record to the ServiceCompletionLog to record the season, start date, number of
  days to complete the service, property ID, employee name, specialization.

  Hint: You will need to create a ServiceCompletionLog to keep track of this data. The procedure will be
  executed after the service has been completed (add the end date to the record). */

CREATE TABLE SERVICECOMPLETIONLOG(
    VisitID      int,
    StartDate    date,
    EndDate      date,
    Season       char(6),
    DaysToFin   varchar(4),
    PropertyID   int,
    EFName       char(30),
    ELName       char(30),
    Specialization tinyint,
    Foreign Key (VisitID) REFERENCES VISITS (VisitID),
    Foreign Key (PropertyID) REFERENCES PROPERTIES (PropertyID));

CREATE PROCEDURE ServiceComp
    @VisitID int
    AS
    BEGIN
        INSERT INTO SERVICECOMPLETIONLOG (VisitID, StartDate, EndDate, Season, DaysToFin, PropertyID,
        EFName, ELName, Specialization)
        VALUES (@VisitID,
        (SELECT StartDate
         FROM VISITS
         WHERE @VisitID = VisitID),
        (SELECT EndDate
         FROM VISITS
         WHERE @VisitID = VisitID),
        (CASE
        WHEN (SELECT EndDate FROM VISITS WHERE @VisitID = VisitID) LIKE '%-03-%'
        OR
        (SELECT EndDate FROM VISITS WHERE @VisitID = VisitID) LIKE '%-04-%'
        OR
        (SELECT EndDate FROM VISITS WHERE @VisitID = VisitID) LIKE '%-05-%'
        THEN 'Spring'
        WHEN (SELECT EndDate FROM VISITS WHERE @VisitID = VisitID) LIKE '%-06-%'
        OR
        (SELECT EndDate FROM VISITS WHERE @VisitID = VisitID) LIKE '%-07-%'
        THEN 'Summer'
        WHEN (SELECT EndDate FROM VISITS WHERE @VisitID = VisitID) LIKE '%-08-%'
        OR
        (SELECT EndDate FROM VISITS WHERE @VisitID = VisitID) LIKE '%-09-%'
        THEN 'Autumn'
        WHEN (SELECT EndDate FROM VISITS WHERE @VisitID = VisitID) LIKE '%-10-%'
        OR
        (SELECT EndDate FROM VISITS WHERE @VisitID = VisitID) LIKE '%-11-%'
        THEN 'Winter'
        ELSE 'Unknown'
        END));
    END;
```

Code part 2

```
        OR
        (SELECT EndDate FROM VISITS WHERE @VisitID = VisitID) LIKE '%-08-%'
        THEN 'Summer'
    WHEN (SELECT EndDate FROM VISITS WHERE @VisitID = VisitID) LIKE '%-09-%'
        OR
        (SELECT EndDate FROM VISITS WHERE @VisitID = VisitID) LIKE '%-10-%'
        OR
        (SELECT EndDate FROM VISITS WHERE @VisitID = VisitID) LIKE '%-11-%'
        THEN 'Fall'
    ELSE 'Winter'
END),
DATEDIFF(day,
        (SELECT StartDate
         FROM VISITS
         WHERE @VisitID = VisitID),
        (SELECT EndDate
         FROM VISITS
         WHERE @VisitID = VisitID)),
(SELECT PropertyID
         FROM VISITS
         WHERE @VisitID = VisitID),
(SELECT EFName
         FROM Employees
         WHERE EmployeeID IN (
             SELECT EmployeeID
             FROM VISITS
             WHERE VisitID = @VisitID)),
(SELECT ELName
         FROM Employees
         WHERE EmployeeID IN (
             SELECT EmployeeID
             FROM VISITS
             WHERE VisitID = @VisitID)),
(SELECT SpecializationID
         FROM Employees
         WHERE EmployeeID IN (
             SELECT EmployeeID
             FROM VISITS
             WHERE VisitID = @VisitID));
END
SELECT *
FROM SERVICECOMPLETIONLOG;
EXEC ServiceComp @VisitID = 26217;
```

| VisitID | StartDate | EndDate | Season | DaysToFin | PropertyID | EFName | ELName | Specialization |
|---------|-----------|---------|--------|-----------|------------|--------|--------|----------------|
| | | | | | | | | |

Result

```
SQLQuery2.sql - ac...UAD\pscheetz (59)* ↗ X
SELECT EmployeeID
      FROM VISITS
        WHERE VisitID = @VisitID)));
END

SELECT *
      FROM SERVICECOMPLETIONLOG;

EXEC ServiceComp @VisitID = 26217;
```

125 %

| | VisitID | StartDate | EndDate | Season | DaysToFin | PropertyID | EFName | ELName | Specialization |
|---|---------|------------|------------|--------|-----------|------------|----------|--------|----------------|
| 1 | 26217 | 2017-05-08 | 2018-08-26 | Summer | 475 | 9355 | Roshelle | Simond | 9 |

Index / Query Plans

Index /Query Plan 1

```
SQLQuery1.sql - ac...AD\bmdavi24 (116)* ↵ X
/* 1. (2) Identify the attributes (at least 2) that should have secondary indexes defined (cluster index is already created) to speed up the query you will define in #3. Take a snapshot of the tables you will use (select * from each table and show the attributes and a few rows of data in each one). */

SELECT *
FROM PROPERTIES;

SELECT *
FROM VISITS;
```

100 %

| | PropertyID | PropName | Address | #Bed | #Bath | SqFt | YrBuilt | RoofType | ExteriorType | ClientID |
|---|------------|----------|---------------------------|------|-------|----------|---------|----------|--------------|----------|
| 1 | 1000 | Subject | 123 Main St | 1 | 1 | 1111.11 | 2019 | tile | brick | 5000000 |
| 2 | 1001 | Bennett | 2946 West Lane | 9 | 9 | 48775.02 | 2006 | concrete | brick | 1000000 |
| 3 | 1106 | Johnson | 8 Kim Pass | 5 | 8 | 49375.71 | 2003 | tile | stucco | 6586146 |
| 4 | 1811 | Melton | 5 Mitchell Park | 3 | 8 | 72935.57 | 1990 | shingles | stucco | 95811088 |
| 5 | 2128 | Mcintyre | 57458 American Ash Street | 9 | 4 | 10811.78 | 1993 | shingles | stucco | 80920878 |
| 6 | 3016 | Klein | 936 Amoth Crossing | 3 | 9 | 16199.14 | 2007 | shingles | stucco | 9038853 |
| 7 | 3190 | Gay | 1 Rieder Crossing | 7 | 4 | 41305.12 | 2012 | concrete | vinyl | 95264136 |
| 8 | 4133 | Cannon | 5 Brown Trail | 5 | 10 | 2533.40 | 2008 | shingles | vinyl | 68603197 |

| | VisitID | DateOfVisit | StartDate | EndDate | Owner | TotalCost | PropertyID | ServiceID | ClientID | EmployeeID |
|---|---------|-------------|------------|------------|------------------|-----------|------------|-----------|----------|------------|
| 1 | 4761 | 2017-04-27 | 2017-06-09 | 2019-03-03 | Brett Randall | 61540.12 | 4270 | 1 | 66233683 | 3492369222 |
| 2 | 8453 | 2017-02-25 | 2017-08-29 | 2018-04-22 | Modesto Oneal | 65027.39 | 6929 | 1 | 47620087 | 4634033099 |
| 3 | 10384 | 2017-01-15 | 2017-11-05 | 2019-07-18 | Lucius Bowers | 90846.03 | 5754 | 1 | 20536872 | 2627959571 |
| 4 | 26217 | 2017-02-25 | 2017-05-08 | 2018-08-26 | Sparkle Grah... | 82603.56 | 9355 | 9 | 45730150 | 929024070 |
| 5 | 28319 | 2017-01-31 | 2017-04-29 | 2020-02-01 | Tyrone Macias | 32810.50 | 8000 | 1 | 8565248 | 1303503027 |
| 6 | 54238 | 2017-02-14 | 2017-10-30 | 2018-06-25 | Shirly Beck | 42291.77 | 7426 | 3 | 36328797 | 4183901477 |
| 7 | 130... | 2017-05-09 | 2017-11-16 | 2019-02-11 | Rigoberto Fin... | 52230.82 | 8901 | 6 | 18896384 | 4063789551 |

Query executed successfully.

Index/Query Plan 2

The screenshot shows a SQL Server Management Studio interface with three main panes:

- Top Pane:** Shows two tabs: "SQLQuery2.sql - a...UAD\bmdavi24 (62))" and "SQLQuery1.sql - ac...AD\bmdavi24 (140))".
- Middle Pane:** Displays the following SQL code:

```
/* 2. (2) Write the SQL commands to create the secondary indexes for the 2 attributes identified in #1. Give reasons for each attribute selected (use comments). Take a snapshot of the SQL code for both indexes. */

CREATE INDEX Index#Bed ON PROPERTIES (#Bed);
/* Attribute selected because we would like to quickly search properties by how many bedrooms they have
because it is one of the most important property attributes for both clients and the company. */

CREATE INDEX IndexVisitDate ON VISITS(DateOfVisit);
/* Attribute selected because we would like to quickly search visits by which date they occurred on
so we can then investigate any issues that may have occurred during a visit on a certain date. */
```

- Bottom Pane:** Shows the "Messages" tab with the message: "Command(s) completed successfully."

Index/Query Plan 3

```
SQLQuery3.sql - a...UAD\bmdavi24 (88)* ▾ X
/* 3. (2) Create an advanced join query and execute (use any multiple tables you wish). Show the query, the data results, the time to complete, and the query plan (take snapshots of each). The idea here is to create a query that will tax the computer resources - force the DBMS to use its resources. */

SET STATISTICS TIME ON;

SELECT CFName, CIPhone, CICity
FROM CLIENTS C
JOIN VISITS V ON V.ClientID = C.ClientID
JOIN PROPERTIES P ON P.PropertyID = V.PropertyID
JOIN SERVICECOMPLETION S ON V.VisitID = S.VisitID
WHERE CFName IN ('Sparkle', 'Modesto', 'Kenyatta', 'Kristopher', 'Fredricka', 'Maxie')
AND #Bed > 3
AND S.PropertyID > #Bath;

/* 4. (2) Optimize & execute the query in #3. Show the query, the data results, the time to complete, and the query plan (take snapshots of each). */

SET STATISTICS TIME ON;

SELECT CFName, CIPhone, CICity
FROM CLIENTS C
WHERE ClientID IN (
```

100 % ▾

| CFName | CIPhone | CICity |
|--------|---------|--------|
|--------|---------|--------|

Results Messages Execution plan

Query executed successfully. | acadsql17.asurite.ad.asu.edu... | ASUAD\bmdavi24 (88) | SP20_CIS365_19668_TEAM2 | 00:00:00 | 0 rows

SQLQuery3.sql - a...UAD\bmdavi24 (88))*

```

/* 3. (2) Create an advanced join query and execute (use any multiple tables you wish). Show the query, the
 data results, the time to complete, and the query plan (take snapshots of each). The idea here is to
 create a query that will tax the computer resources - force the DBMS to use its resources. */

SET STATISTICS TIME ON;

SELECT CFName, ClPhone, ClCity
  FROM CLIENTS C
 JOIN VISITS V ON V.ClientID = C.ClientID
 JOIN PROPERTIES P ON P.PropertyID = V.PropertyID
 JOIN SERVICECOMPLETION S ON V.VisitID = S.VisitID
 WHERE CFName IN ('Sparkle', 'Modesto', 'Kenyatta', 'Kristopher', 'Fredricka', 'Maxie')
   AND #Bed > 3
   AND S.PropertyID > #Bath;

/* 4. (2) Optimize & execute the query in #3. Show the query, the data results, the time to complete, and
 the query plan (take snapshots of each). */

SET STATISTICS TIME ON;

SELECT CFName, ClPhone, ClCity
  FROM CLIENTS C
 WHERE ClientID IN (

```

100 %

Results Messages Execution plan

SQL Server parse and compile time:
CPU time = 0 ms, elapsed time = 0 ms.

SQL Server Execution Times:
CPU time = 0 ms, elapsed time = 0 ms.

SQL Server parse and compile time:
CPU time = 16 ms, elapsed time = 77 ms.

(0 row(s) affected)

(1 row(s) affected)

SQL Server Execution Times:
CPU time = 0 ms, elapsed time = 60 ms.

SQL Server parse and compile time:
CPU time = 0 ms, elapsed time = 0 ms.

SQL Server Execution Times:
CPU time = 0 ms, elapsed time = 0 ms.

100 %

Query executed successfully. | acadsql17.asurite.ad.asu.edu | ASUAD\bmdavi24 (88) | SP20_CIS365_19668_TEAM2 | 00:00:00 | 0 rows

SQLQuery3.sql - a...UAD\bmdavi24 (88) * X

```

/*
3. (2) Create an advanced join query and execute (use any multiple tables you wish). Show the query, the
data results, the time to complete, and the query plan (take snapshots of each). The idea here is to
create a query that will tax the computer resources - force the DBMS to use its resources.
*/

SET STATISTICS TIME ON;

SELECT CFName, ClPhone, ClCity
FROM CLIENTS C
JOIN VISITS V ON V.ClientID = C.ClientID
JOIN PROPERTIES P ON P.PropertyID = V.PropertyID
JOIN SERVICECOMPLETION S ON V.VisitID = S.VisitID
WHERE CFName IN ('Sparkle', 'Modesto', 'Kenyatta', 'Kristopher', 'Fredricka', 'Maxie')
AND #Bed > 3

```

100 % ▶

Results Messages Execution plan

Query 1: Query cost (relative to the batch): 100%

```

SELECT CFName, ClPhone, ClCity FROM CLIENTS C JOIN VISITS V ON V.ClientID = C.ClientID JOIN PROPERTIES P ON P.Prope...

```

The execution plan diagram illustrates the query's execution flow. It starts with a 'Nested Loops (Inner Join)' node for the join between 'CLIENTS' and 'VISITS'. This is followed by another 'Nested Loops (Inner Join)' node for the join between 'VISITS' and 'PROPERTIES'. A third 'Nested Loops (Inner Join)' node joins 'PROPERTIES' with 'SERVICECOMPLETION'. Finally, a 'Table [VISITS]' node is shown, connected to a 'Clustered Index Seek (Clustered)' node for the 'VISITS' table. The total cost for the query is 100%.

< | > +

Query executed successfully. acadsq17.asurite.ad.asu.edu... | ASUAD\bmdavi24 (88) | SP20_CIS365_19668_TEAM2 | 00:00:00 | 0 rows

SQLQuery3.sql - a...UAD\bmdavi24 (88) * X

```

/*
3. (2) Create an advanced join query and execute (use any multiple tables you wish). Show the query, the
data results, the time to complete, and the query plan (take snapshots of each). The idea here is to
create a query that will tax the computer resources - force the DBMS to use its resources. */

SET STATISTICS TIME ON;

SELECT CFName, ClPhone, ClCity
FROM CLIENTS C
JOIN VISITS V ON V.ClientID = C.ClientID
JOIN PROPERTIES P ON P.PropertyID = V.PropertyID
JOIN SERVICECOMPLETION S ON V.VisitID = S.VisitID
WHERE CFName IN ('Sparkle', 'Modesto', 'Kenyatta', 'Kristopher', 'Fredricka', 'Maxie')
AND #Bed > 3

```

100 %

Results Messages Execution plan

Query 1: Query cost (relative to the batch): 100%

SELECT CFName, ClPhone, ClCity FROM CLIENTS C JOIN VISITS V ON V.ClientID = C.ClientID JOIN PROPERTIES P ON P.Prope...

```

graph TD
    A[Client Loops  
Inner Join  
Cost: 0 %] --> B[Nested Loops  
(Inner Join)  
Cost: 0 %]
    B --> C[Nested Loops  
(Inner Join)  
Cost: 0 %]
    C --> D[Table Scan  
[SERVICECOMPLETION] [S]  
Cost: 25 %]
    D --> E[Clustered Index Seek (Clustered)  
[VISITS].[PK_VISITS_4D3AA1BE09687...  
Cost: 25 %]
    E --> F[Clustered Index Seek (Clustered)  
[CLIENTS].[PK_CLIENTS_E67E1A044EE...  
Cost: 25 %]
    F --> G[Clustered Index Seek (Clustered)  
[PROPERTIES].[PK_PROPERTY_70C9A75...  
Cost: 25 %]

```

< > +

Query executed successfully. | acadsql17.asurite.ad.asu.edu... | ASUAD\bmdavi24 (88) | SP20_CIS365_19668_TEAM2 | 00:00:00 | 0 rows

Indexes/Query Plan 4 (including Optimization)

SQLQuery3.sql - a...UAD\bmdavi24 (88)* ▾ X

```

/* 4. (2) Optimize & execute the query in #3. Show the query, the data results, the time to complete, and
the query plan (take snapshots of each). */

SET STATISTICS TIME ON;

SELECT CFName, ClPhone, ClCity
FROM CLIENTS C
WHERE ClientID IN (
    SELECT ClientID
    FROM VISITS --JOINING VISITS TABLE BY CONNECTING V.ClientID = C.ClientID
    WHERE CFName IN ('Sparkle', 'Modesto', 'Kenyatta', 'Kristopher', 'Fredricka', 'Maxie') --RESTRAINT FROM ORIGINAL QUERY
    AND PropertyID IN (
        SELECT PropertyID
        FROM PROPERTIES --JOINING PROPERTIES TABLE BY CONNECTING P.PropertyID = V.PropertyID
        WHERE #Bed > 3 --RESTRAINT FROM ORIGINAL QUERY
        AND PropertyID IN (
            SELECT PropertyID
            FROM SERVICECOMPLETION --JOINING SERVICECOMPLETION TABLE VIA V.VisitID = S.VisitID
            WHERE PropertyID > #Bath)); --RESTRAINT FROM ORIGINAL QUERY

```

100 % ▾

Results Messages Execution plan

| CFName | ClPhone | ClCity |
|--------|---------|--------|
| | | |

SQLQuery3.sql - a...UAD\bmdavi24 (88)* ▾ X

```

/* 4. (2) Optimize & execute the query in #3. Show the query, the data results, the time to complete, and
the query plan (take snapshots of each). */

SET STATISTICS TIME ON;

SELECT CFName, ClPhone, ClCity
FROM CLIENTS C
WHERE ClientID IN (
    SELECT ClientID
    FROM VISITS --JOINING VISITS TABLE BY CONNECTING V.ClientID = C.ClientID
    WHERE CFName IN ('Sparkle', 'Modesto', 'Kenyatta', 'Kristopher', 'Fredricka', 'Maxie') --RESTRAINT FROM ORIGINAL QUERY
    AND PropertyID IN (
        SELECT PropertyID
        FROM PROPERTIES --JOINING PROPERTIES TABLE BY CONNECTING P.PropertyID = V.PropertyID
        WHERE #Bed > 3 --RESTRAINT FROM ORIGINAL QUERY
        AND PropertyID IN (
            SELECT PropertyID
            FROM SERVICECOMPLETION --JOINING SERVICECOMPLETION TABLE VIA V.VisitID = S.VisitID
            WHERE PropertyID > #Bath)); --RESTRAINT FROM ORIGINAL QUERY

```

100 % ▾

Results Messages Execution plan

SQL Server parse and compile time:
CPU time = 0 ms, elapsed time = 0 ms.

SQL Server Execution Times:
CPU time = 0 ms, elapsed time = 0 ms.

SQL Server parse and compile time:
CPU time = 16 ms, elapsed time = 19 ms.

(0 row(s) affected)

(1 row(s) affected)

SQL Server Execution Times:
CPU time = 0 ms, elapsed time = 22 ms.

SQL Server parse and compile time:
CPU time = 0 ms, elapsed time = 0 ms.

SQL Server Execution Times:
CPU time = 0 ms, elapsed time = 0 ms.

|

100 % ▾

Query executed successfully. | acadsql17.asurite.ad.asu.edu | ASUAD\bmdavi24 (88) | SP20_CIS365_19668_TEAM2 | 00:00:00 | 0 rows

SQlQuery3.sql - a...UAD\bmdavi24 (88)*

```

/* 4. (2) Optimize & execute the query in #3. Show the query, the data results, the time to complete, and the query plan (take snapshots of each). */

SET STATISTICS TIME ON;

SELECT CFName, ClPhone, ClCity
FROM CLIENTS C
WHERE ClientID IN (
    SELECT ClientID
    FROM VISITS --JOINING VISITS TABLE BY CONNECTING V.ClientID = C.ClientID
    WHERE CFName IN ('Sparkle','Modesto','Kenyatta','Kristopher','Fredricka','Maxie') --RESTRAINT FROM ORIGINAL QUERY
    AND PropertyID = 1
)

```

100 %

Results Messages Execution plan

Query 1: Query cost (relative to the batch): 100%

SELECT CFName, ClPhone, ClCity FROM CLIENTS C WHERE ClientID IN (SELECT ClientID FROM VISITS --JOINING VISITS TABL..

```

graph TD
    A[SELECT CFName, ClPhone, ClCity] --> B[Nested Loops (Left Semi Join)]
    B --> C[Clustered Index Scan (Clustered) [CLIENTS].[PK_CLIENTS_E67E1A044EE... Cost: 16%]
    C --> D[Nested Loops (Left Semi Join) Cost: 0%]
    D --> E[Nested Loops (Inner Join) Cost: 0%]
    E --> F[Index Seek (NonClustered) [VISITS].[UQ_VISITS_E67E1A0575F54... Cost: 20%]
    F --> G[Key Lookup (Clustered) [VISITS].[PK_VISITS_4D3AA1BE09687... Cost: 23%]
    G --> H[Nested Loops (Left Semi Join) Cost: 0%]
    H --> I[Clustered Index Seek (Clustered) [PROPERTIES].[PK_PROPERTIES_70C5A... Cost: 23%]
    I --> J[Table Scan [SERVICECOMPLETION] Cost: 18%]

```

< | > +

Query executed successfully.

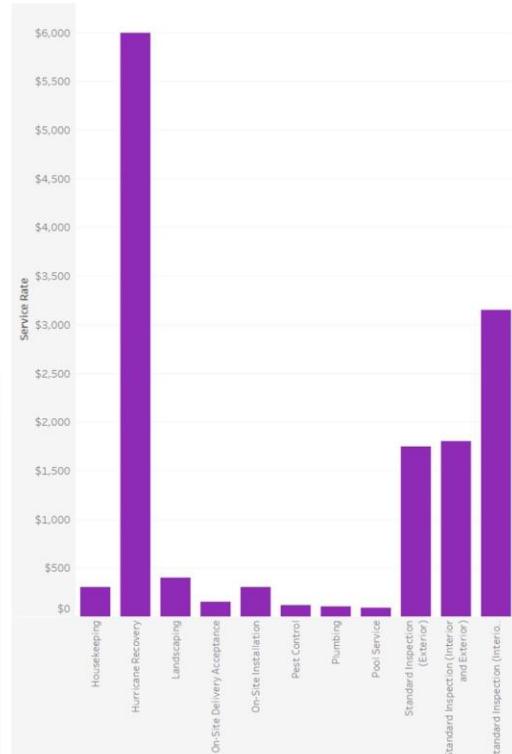
acadsql17.asurite.ad.asu.edu | ASUAD\bmdavi24 (88) | SP20_CIS365_19668_TEAM2 | 00:00:00 | 0 rows

Analytical Dashboard / Visualizations

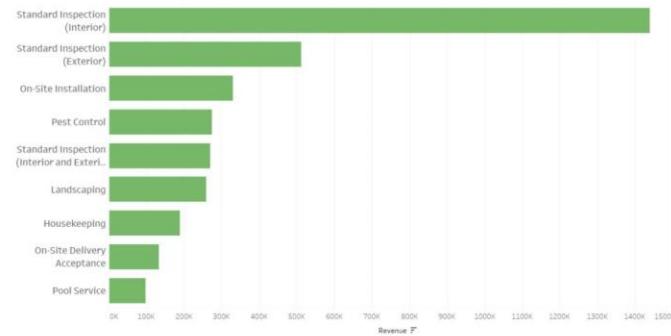
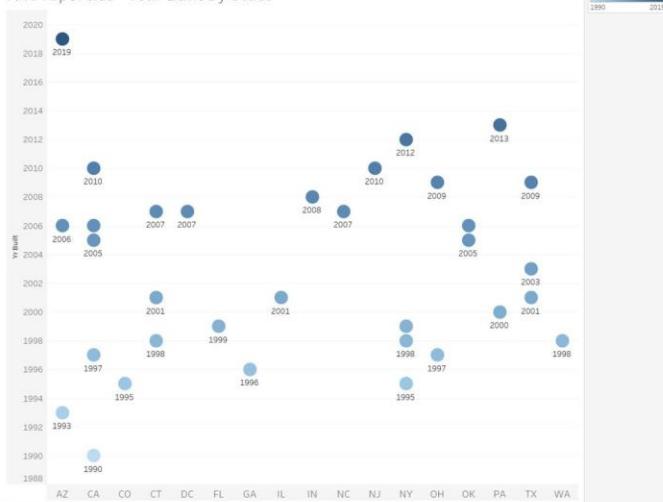
Number of Employees by Specialization

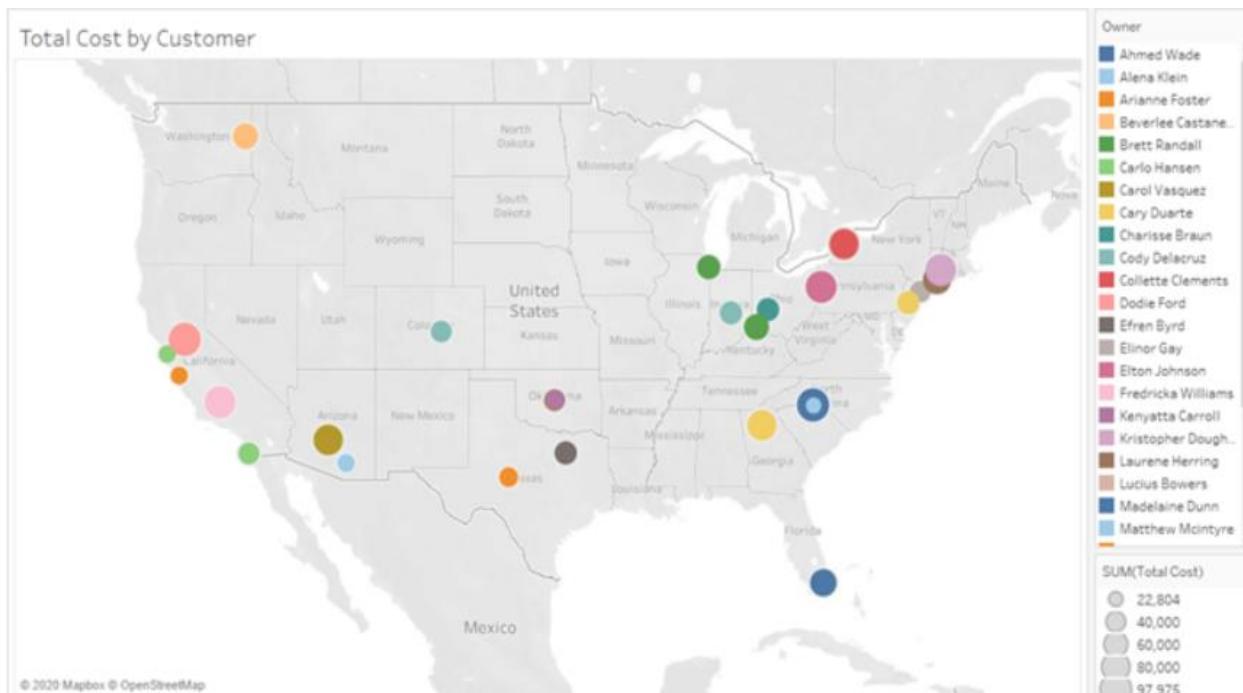


Total Service Revenue

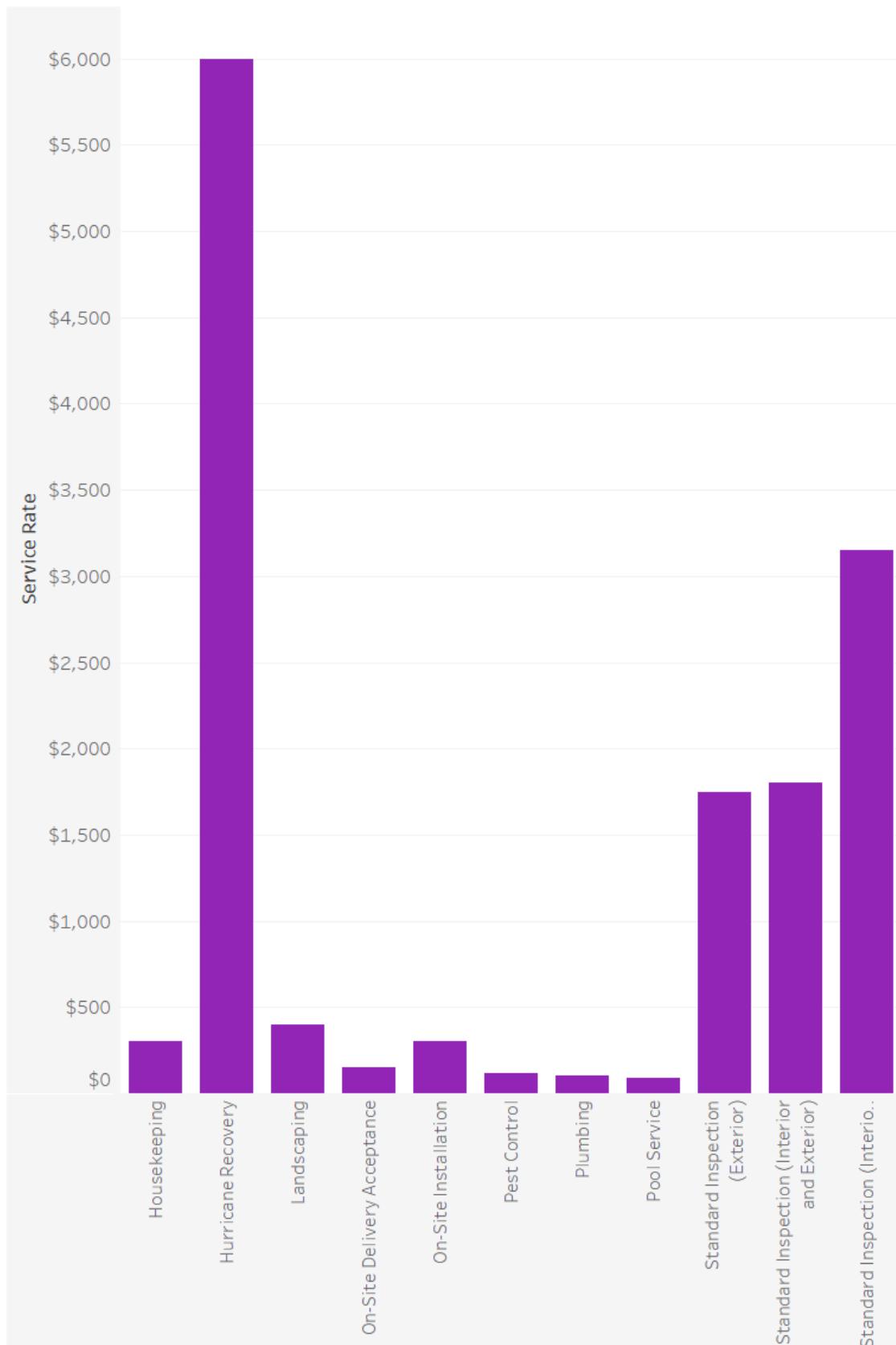


All Properties - Year Built by State

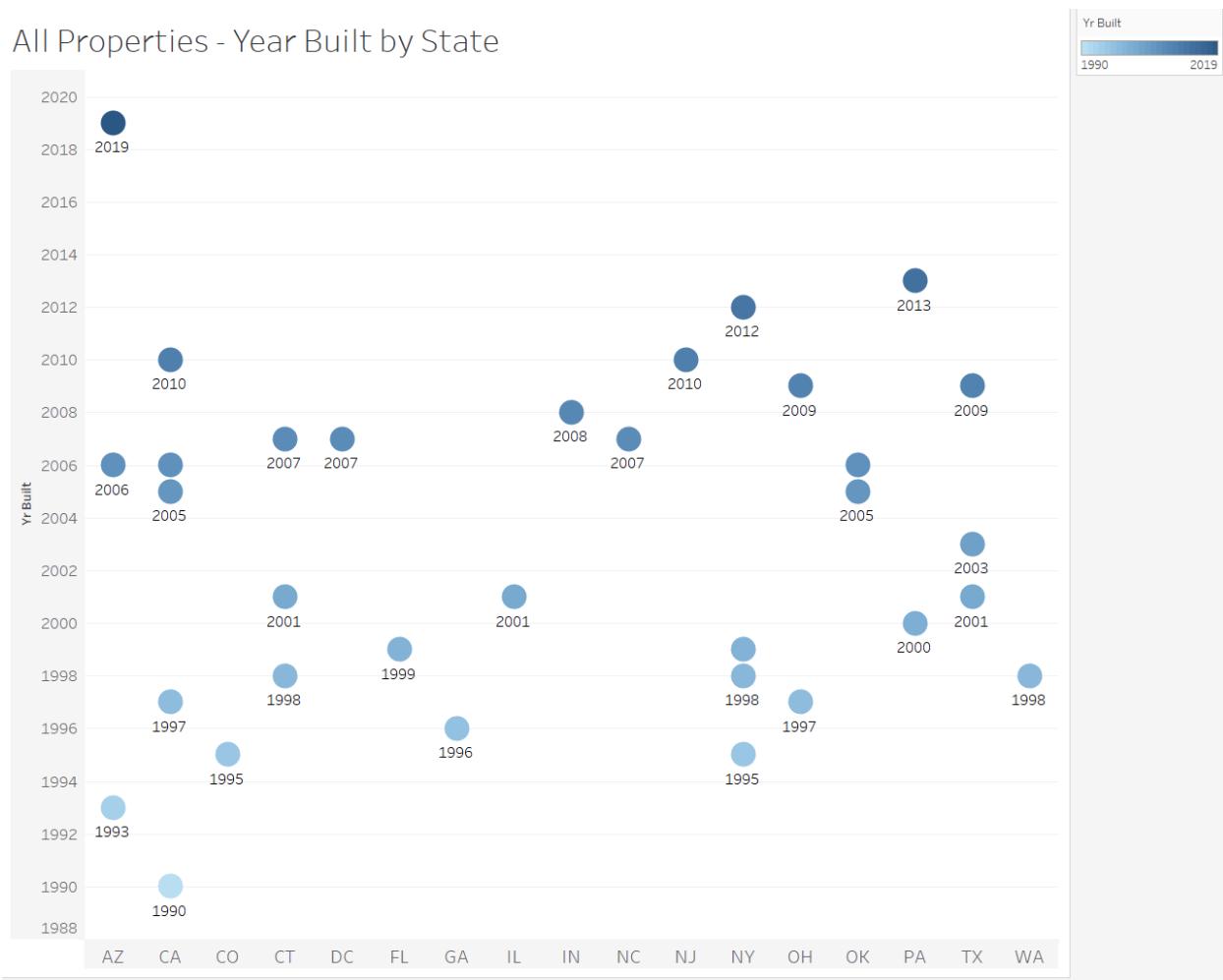




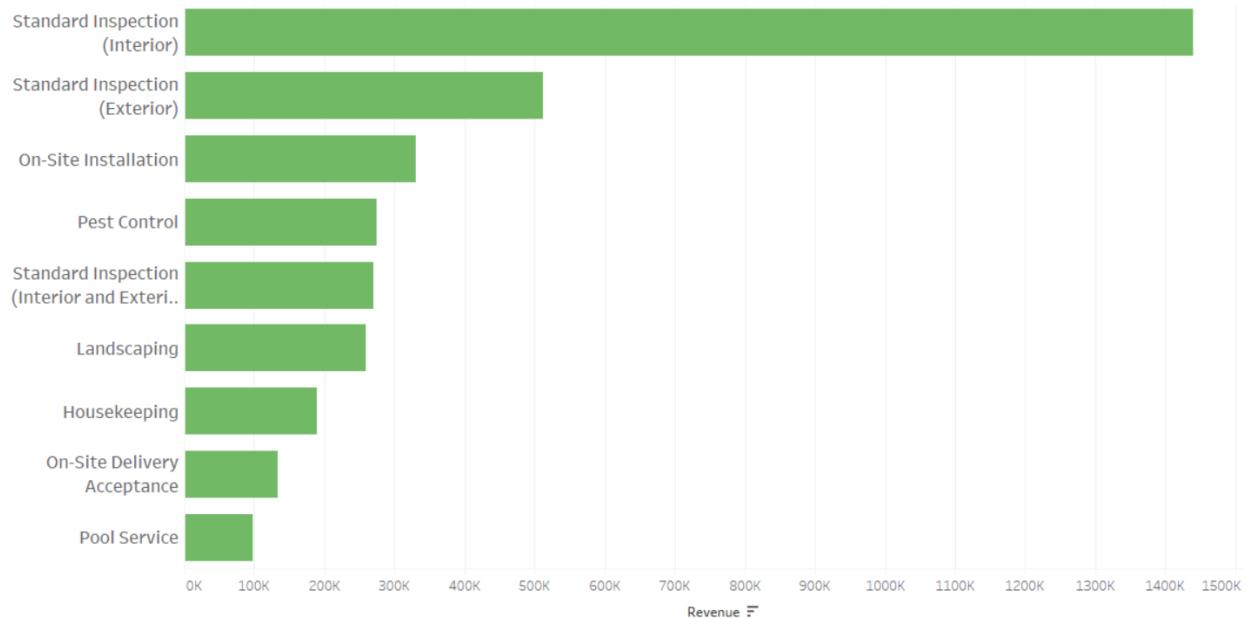
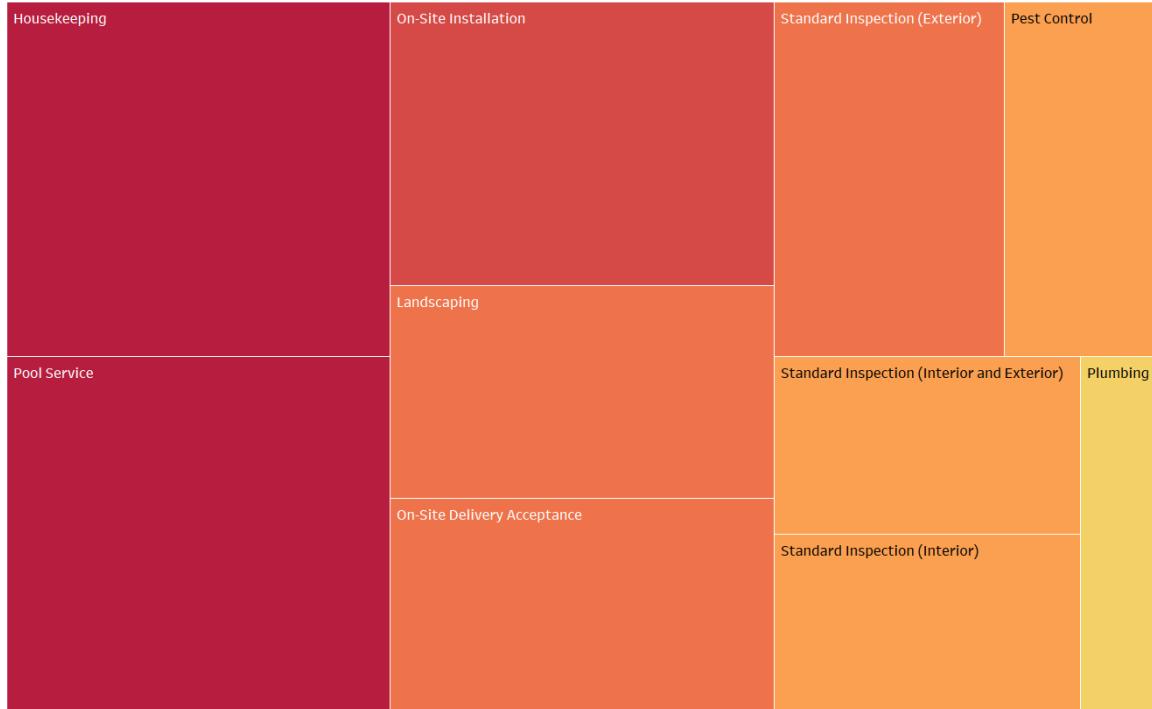
Total Service Revenue



All Properties - Year Built by State



Number of Employees by Specialization



Appendix A: SQL Scripts

SQL Scripts of Database Creation

```
/* CREATING Employees Table and Inserting Employee Data */

CREATE TABLE EMPLOYEES (
    EmployeeID varchar(10)  Not Null Unique,
    EFName      Char(30)    Not Null,
    ELName      Char(30)    Not Null,
    HourlyRate  Decimal(5,2) Not Null,
    EAddress    Char(25)    Not Null,
    EPhone      Varchar(10) Not Null,
    SpecializationID Tinyint          Not Null,
    State       Char(30)    Not Null,
    City        Char(30)    Not Null,
    Primary Key (EmployeeID))

INSERT into employees values (5840844404,      'Nan
ette', 'Girardengo',      19.10, '2150 Surrey Parkway', 3613762359, 17, 'TX',
'Corpus Christi')
INSERT into employees values (4063789551,      'Albrecht', 'Marcum',
71.09, '971 Oakridge Drive', 3109325900, 10, 'CA', 'Santa Monica')
INSERT into employees values (2075868651,      'Mallorie', 'Ernshaw',
77.05, '3407 Stoughton Road', 9162541042, 21, 'CA', 'Sacramento')
INSERT into employees values (2627959571,      'Susi',      'Sharvill',
37.13, '466 Sundown Pass', 9167692822, 25, 'CA', 'Sacramento')
insert into employees values (6975164124,      'Rob',      'Rolston', 83.64, '8
Arkansas Lane', 5611606254, 26, 'FL', 'Boca Raton')
insert into employees values (251747956,      'Aubine',   'Berndtsson',
85.97, '250 Linden Trail', 5023814520, 15, 'KY', 'Louisville')
insert into employees values (7895046015,      'Dannel',   'Oyley',
86.13, '1372 Warrior Circle', 2025327659, 16, 'DC', 'Washington')
insert into employees values (7583088441,      'Gilberta', 'Way', 91.38,
'0094 Fair Oaks Pass', 3219209624, 13, 'FL', 'Melbourne')
insert into employees values (6918216884,      'Gabbie',   'Asquith',
38.68, '040 Eagan Trail', 3129173112, 28, 'IL', 'Chicago')
insert into employees values (4634033099,      'Emanuel',  'Cornelius',
91.92, '51 Little Fleur Park', 3133227214, 26, 'MI', 'Detroit')
```

```

insert into employees values (7601577954, 'Dillon' , 'Jillettt'
,86.55, '9 Darwin Lane', 2601125755, 28 , 'IN', 'Fort Wayne')
insert into employees values (1303503027, 'Ulrikaumeko', 'De
Dantesie', 32.32, '456 Killdeer Alley', 3235467518, 29, 'CA', 'Los
Angeles')
insert into employees values (52765873, 'Larina', 'Ritter',
57.68, '817 Mcguire Point' , 3035498712, 14, 'CO', 'Denver')
insert into employees values (572549237 , 'Ree', 'Pauler',
54.52, '63971 Ohio Parkway', 6031494909, 25 , 'NH', 'Manchester')
insert into employees values (6011962951, 'Arty', 'Carlsson',
46.69, '4273 Jenna Point', 5139756243, 18 , 'OH' , 'Cincinnati')
insert into employees values (362079200, 'York', 'Curgenuer'
,88.84, '486 Lake View Terrace', 5402905768, 21 , 'VA'
,'Roanoke')
insert into employees values (5436655387, 'Caye', 'Hamnett', 24.48,
'9217 Dixon Junction', 5124385325, 22 , 'TX', 'Austin')
insert into employees values (2985053385, 'Danica', 'Kensley',
88.88, '788 Havey Court', 6261594582 ,21, 'CA', 'Alhambra')
insert into employees values (9918294620, 'Rollin', 'Rollins',
81.27, '38831 Michigan Lane', 7132344169, 30 , 'TX' , 'Humble')
insert into employees values (4183901477, 'Ker' , 'Rebeck', 77.43,
'0088 Warner Drive', 7182926228, 17, 'NY', 'New York City')
insert into employees values (5966772740, 'Kim', 'Gagen', 99.6,
'7012 Cambridge Parkway', 7279903384, 15, 'FL', 'Clearwater')
insert into employees values (6626125260, 'Ilyssa', 'Mold',
59.62, '67741 Gateway Parkway' ,4804043193, 16 , 'AZ', 'Phoenix')
insert into employees values (3492369222, 'Winifield' , 'Winskill'
,47.14, '97 3rd Pass', 5152085062, 25, 'IA', 'Des Moines')
insert into employees values (5804237177, 'Leola', 'Gannan',
22.16, '05731 Lindbergh Junction', 2109384822 ,27 , 'TX', 'San
Antonio')
insert into employees values (977324890, 'Sheffy', 'Hinkley',
73.26, '20961 Aberg Way', 2062857615, 29, 'WA', 'Seattle')
insert into employees values (229054573, 'Clarabelle', 'Tearle',
82.16, '69902 Dorton Junction', 7166276139, 22, 'NY', 'Buffalo')
insert into employees values (5864813592, 'Rorie', 'Betancourt',
33.58, '27440 Caliangt Way', 4154904618, 28, 'CA', 'San Francisco')
insert into employees values (8187560116, 'Sherrie', 'McKiddin',
85.79, '6122 Bonner Street', 6264824160, 15 , 'CA', 'Los Angeles')
insert into employees values (5477528289, 'Durante', 'Dreier',
76.82, '73 Warbler Drive', 5134194849, 17 , 'OH', 'Cincinnati')
insert into employees values (929024070, 'Roshelle', 'Simond',
58.03, '4 Eggendart Circle', 9175274534, 19, 'NY', 'Jamaica')

select * from employees;

```

```

/* Creating SERVICES Table and Inserting SERVICES Data */

CREATE TABLE SERVICES (
    ServiceID      Int          Not Null Unique,
    ServiceDesc    Char(255)    Not Null,
    ServiceRate    Decimal(6,2) Not Null,
    Primary Key (ServiceID));

    insert into services values (1,'Standard Inspection (Interior)',      350)
    insert into services values (2,'Standard Inspection (Exterior)',      350)
    insert into services values (3,'Standard Inspection (Interior and
Exterior)',   600)
    insert into services values (4,'On-Site Delivery Acceptance',  75)
    insert into services values (5,'On-Site Installation',     150)
    insert into services values (6,'Housekeeping',        150)
    insert into services values (7,'Pest Control',         60)
    insert into services values (8,'Pool Service'        ,90)
    insert into services values (9,'Landscaping',        200)

    select * from services;

/* Creating CLIENTS Table and Inserting CLIENTS Data */

CREATE TABLE CLIENTS (
    ClientID integer not null unique,
    ClName char(30) not null,
    ClAddress char(30) not null,
    ClPhone varchar(10) not null,
    YearsServiced integer not null,
    ClEmail char(30) not null,
    ClState char(2) not null,
    ClCity char(30) not null,
    Primary Key(ClientID));

    insert into clients values (47620087, 'Modesto Oneal',    '15 Walton
Pass', 8608999058, 13.86, 'hrenton0@ted.com',      'CT', 'Hartford')
    insert into clients values (61406868, 'Fredricka Williams',    '2012
Blackbird Junction', 8059920105, 22.73, 'pbehnecke1@znet.com'  , 'CA',
'Bakersfield')
    insert into clients values (8565248, 'Tyrone Macias',    '35 Tony
Avenue', 2127324383, 12.44, 'mshellum2@marketwatch.com','NY',      'New York
City')
    insert into clients values (9038853, 'Alena Klein',       '936 Amoth

```

```

Crossing', 7042993362, 23.45,'ssturr3@tumblr.com', 'NC', 'Charlotte')
    insert into clients values (81164013, 'Arianne Foster', '53675 Alpine
Street', 8318228004, 13.22,'rexposito4@ucoz.com', 'CA', 'Salinas')
    insert into clients values (68603197, 'Vernon Cannon', '5 Brown Trail',
3177883221, 6.27, 'mchiommienti5@usgs.gov', 'IN', 'Indianapolis')
    insert into clients values (91761429, 'Cody Delacruz', '3 Myrtle Park'
,7192665655, 7.82, 'lstrowan6@odnoklassniki.ru', 'CO', 'Colorado
Springs')
    insert into clients values (55034572, 'Kristopher Dougherty', '1 Dexter
Crossing', 8603854839, 20.49,'epudan7@example.com', 'CT', 'Hartford')
    insert into clients values (83696889, 'Ahmed Wade', '09320 Hansons
Place', 7868831170, 11.86,'ghaughton8@ebay.com', 'FL', 'Miami')
    insert into clients values (7101753, 'Elton Johnson', '4838 Lakeland
Terrace', 7245078034, 5.07, 'cskatcher9@behance.net', 'PA', 'New
Castle')
    insert into clients values (45730150, 'Sparkle Graham', '453 Miller
Crossing', 7065809309, 20.49,'lmewitta@meetup.com', 'GA', 'Cumming')
    insert into clients values (36328797, 'Shirly Beck', '2946 West
Lane', 4058676001, 23.5, 'salesb@wikispaces.com', 'OK', 'Oklahoma City')
    insert into clients values (3496507, 'Carlo Hansen', '4 Loomis
Point', 4159574029, 17.41,'vjakubowicz@howstuffworks.com', 'CA', 'San
Francisco')
    insert into clients values (8977331, 'Beverlee Castaneda', '17912
Farmco Parkway', 5097900342, 23.43, 'hswabyd@cdc.gov', 'WA', 'Spokane')
    insert into clients values (66233683, 'Brett Randall', '902 Lotheville
Road', 5131931977, 21.25, 'tgodlontone@networkchange.org', 'OH',
'Cincinnati')
    insert into clients values (64593989, 'Maxie Daugherty', '17 Fulton
Road', 3257507833, 7.8, 'tgoodanewf@networkchange.org', 'TX', 'San Angelo')
    insert into clients values (93971647, 'Madelaine Dunn', '4 Eagan
Avenue', 7049563884, 9.61, 'yohareg@fda.gov', 'NC', 'Charlotte')
    insert into clients values (65321745, 'Laurene Herring', '75 Amoth
Avenue', 2036971420, 14.46,'bjardeinh@cnn.com', 'CT', 'New Haven')
    insert into clients values (79147483, 'Cary Duarte', '38294 Maple
Parkway', 6096763322, 20.1, 'jhaythi@imageshack.us', 'NJ', 'Trenton')
    insert into clients values (95811088, 'Rodrigo Melton', '5 Mitchell
Park', 8586468390, 12.56, 'ewalasikj@1688.com', 'CA', 'San Diego')
    insert into clients values (95264136, 'Elinor Gay', '1 Rieder
Crossing', 2125065536, 15.34, 'sweatherburnk@tripod.com', 'NY', 'New York
City')
    insert into clients values (83443310, 'Dodie Ford', '7762 Cottonwood
Point', 9164703118, 4.7, 'pwallbuttonl@de.vu', 'CA', 'Sacramento')
    insert into clients values (18896384, 'Rigoberto Finley', '266 Weeping
Birch Circle', 3121237328, 17.74, 'smurthwaitem@blinklist.com', 'IL',
'Chicago')

```

```

        insert into clients values (3548957, 'Carol Vasquez', '9378 Sugar
Way', 6024770379, 16.06, 'zspanswickn@twitpic.com', 'AZ', 'Phoenix')
        insert into clients values (20536872, 'Lucius Bowers' , '441 South
Avenue', 7162972323, 24.68, 'mcantero@loc.gov', 'NY', 'Buffalo')
        insert into clients values (80920878, 'Matthew McIntyre', '57458
American Ash Street', 5208881919, 13.71, 'kmowsdillp@phoca.cz', 'AZ',
'Tucson')
        insert into clients values (39862515, 'Charisse Braun', '1220 8th
Plaza', 9379393341, 15.37, 'sboshardq@wikispaces.com' , 'OH',
'Springfield')
        insert into clients values (50655329, 'Kenyatta Carroll', '2979
Sundown Terrace', 4056290290, 14.98, 'sjulyanr@howstuffworks.com', 'OK',
'Oklahoma City')
        insert into clients values (4812308, 'Efren Byrd', '148 Transport
Parkway', 2146294515, 19.28, 'cdunstans@wikipedia.org', 'TX', 'Dallas')
        insert into clients values (87915581 , 'Collette Clements', '26
Lakeland Way', 7164958639, 9.39, 'ljjorczykt@pcworld.com' , 'NY', 'Buffalo')
        insert into clients values (6586146, 'Carolin Johnson', '8 Kim Pass',
2813112153 , 2.41 , 'measonu@shareasale.com', 'TX', 'Houston')
        insert into clients values (68203269, 'Erika Andrews', '229 David
Point', 2029142772, 22.84, 'rbainesv@npr.org', 'DC', 'Washington')

select * from clients;

/* Creating PROPERTIES Table and Inserting PROPERTIES Data */

CREATE TABLE PROPERTIES (
    PropertyID      integer      Not Null Unique,
    PropName        Char(30)     Not Null,
    Address         Char(255)    Not Null,
    #Bed            integer      Not Null,
    #Bath           integer      Not Null,
    SqFt            Decimal(8,2) Not Null,
    YrBuilt          int        Not Null,
    RoofType        Char(20)     Not Null,
    ExteriorType   Char(20)     Not Null,
    ClientID integer not null unique,
    Foreign Key (ClientID) REFERENCES CLIENTS (ClientID));

    insert into properties values (6929, 'Oneal'      , '15 Walton Pass', 9
,6, 47189.29, 2001, 'tile','brick', 47620087)
    insert into properties values (4233, 'Williams', '2012 Blackbird
Junction', 10, 1, 34531.05, 2006, 'concrete', 'vinyl', 61406868)

```

```

insert into properties values (8000, 'Macias',      '35 Tony Avenue',  5,
9,    7987.77      ,1995,'concrete',   'stucco',     8565248)
insert into properties values (3016, 'Klein',       '936 Amoth Crossing',
3,    9,    16199.14,  2007, 'shingles',  'stucco',     9038853)
insert into properties values (7120, 'Foster',      '53675 Alpine Street',
7,7,  22937.05,  2005, 'shingles',  'brick'        ,81164013)
insert into properties values (4133, 'Cannon',      '5 Brown Trail',   5,
10,   2533.4,    2008, 'shingles',  'vinyl',       68603197)
insert into properties values (6434, 'Delacruz',    '3 Myrtle Park',  5,
9,    6046.6,    1995, 'tile',       'brick',       91761429)
insert into properties values (4630, 'Dougherty',   '1 Dexter Crossing',
9,    3,    49104.89, 1998, 'tile',       'vinyl',     55034572)
insert into properties values (9350, 'Wade',        '09320 Hansons Place',
4,    10,   92979.52, 1999, 'shingles',  'brick',     83696889)
insert into properties values (5555, 'Johnson',    '4838 Lakeland
Terrace', 5,    3,    76643.08, 2013, 'concrete',  'stucco',     7101753)
insert into properties values (9355, 'Graham',     '453 Miller Crossing',
2,    6,    17395.04, 1996, 'tile',       'stucco',     45730150)
insert into properties values (7426, 'Beck',        '2946 West Lane',  9,
9,    48775.02,  2006, 'concrete',  'brick',       36328797)
insert into properties values (7388, 'Hansen',     '4 Loomis Point',  3,
10,   64608.85, 2010, 'tile',       'stucco',     3496507)
insert into properties values (8123, 'Castaneda',   '17912 Farmco Parkway',
5,    2,    50427.55, 1998, 'tile',       'vinyl',     8977331)
insert into properties values (4270, 'Randall',    '902 Lotheville Road',
8,    9,    55927.76, 1997, 'tile',       'stucco',     66233683)
insert into properties values (8448, 'Daugherty',  '17 Fulton Road',  3,
7,    32394.89,  2009, 'shingles',  'stucco',     64593989)
insert into properties values (4879, 'Dunn',        '4 Eagan Avenue',  6,
4,    8271.6,   2007, 'tile',       'stucco',     93971647)
insert into properties values (4855, 'Herring',    '75 Amoth Avenue', 7,
9,    92573.03, 2007, 'tile',       'vinyl',      65321745)
insert into properties values (5726, 'Duarte',     '38294 Maple Parkway',
8,    9,    76918, 2010, 'concrete',  'brick',      79147483)
insert into properties values (1811, 'Melton',     '5 Mitchell Park', 3,
8,    72935.57, 1990, 'shingles',  'stucco',     95811088)
insert into properties values (3190, 'Gay',         '1 Rieder Crossing', 7,
4,    41305.12, 2012, 'concrete',  'vinyl',      95264136)
insert into properties values (8770, 'Ford',        '7762 Cottonwood
Point', 2,    7,    70034.54, 1997, 'concrete',  'brick',     83443310)
insert into properties values (8901, 'Finley',      '266 Weeping Birch
Circle', 8,    10,   33835.04, 2001, 'concrete',  'stucco',     18896384)
insert into properties values (5882, 'Vasquez',    '9378 Sugar Way',  9,
3,    90228.25, 2006, 'concrete',  'stucco',     3548957)
insert into properties values (5754, 'Bowers',     '441 South Avenue',

```

```

2,    7,    66034.28,    1998, 'shingles', 'brick',    20536872)
insert into properties values (2128, 'Mcintyre', '57458 American Ash
Street', 9,    4,    10811.78,    1993, 'shingles', 'stucco',    80920878)
insert into properties values (9051, 'Braun',    '1220 8th Plaza', 3,
10,    48656.39,    2009, 'tile',    'stucco',    39862515)
insert into properties values (7059, 'Carroll',   '2979 Sundown Terrace',
9,    10,    30238.21,    2005, 'shingles', 'stucco',    50655329)
insert into properties values (6717, 'Byrd',    '148 Transport
Parkway', 9,    8,    82821.34,    2001, 'concrete', 'brick',    4812308)
insert into properties values (9851, 'Clements',  '26 Lakeland Way', 9,
8,    40206.13,    1999, 'concrete', 'stucco',    87915581)
insert into properties values (1106, 'Johnson',  '8 Kim Pass', 5,
8,    49375.71,    2003, 'tile','stucco',    6586146)
insert into properties values (6789, 'Andrews',  '229 David Point', 6,
1,    38233.35,    2007, 'shingles', 'vinyl',    68203269)

select * from properties;

/* Creating VISITS Table and Inserting VISITS Data */

CREATE TABLE VISITS (
VisitID          Int      Not Null Unique,
DateOfVisit      Date     Not Null,
StartDate        Date     Not Null,
EndDate          Date     Not Null,
Owner            Char(255) Not Null,
TotalCost        Decimal(7,2) Not Null,
PropertyID       integer  Not Null Unique,
ServiceID        Int      Not Null,
ClientID         integer not null unique,
EmployeeID       varchar(10) Not Null Unique,
Foreign Key (PropertyID) REFERENCES PROPERTIES (PropertyID),
Foreign Key (ServiceID) REFERENCES SERVICES (ServiceID),
Foreign Key (ClientID) REFERENCES CLIENTS (ClientID),
Foreign Key (EmployeeID) REFERENCES EMPLOYEES (EmployeeID)

insert into visits values (8453,      '2017-02-25',      '2017-08-29',
'2018-04-22',      'Modesto Oneal',    65027.39,    6929, 1,    47620087,
4634033099)
insert into visits values (150336,     '2017-04-01',     '2017-12-01',
'2019-01-25',     'Fredricka Williams', 90087.57,    4233, 1,

```

```

61406868,    7601577954)
insert into visits values (28319,      '2017-01-31',      '2017-04-29',
'2020-02-01',      'Tyrone Macias', 32810.5,     8000, 1      ,8565248,
1303503027)
insert into visits values (489302,     '2017-04-11',      '2017-08-28',
'2018-09-02',      'Alena Klein',   22803.75,    3016, 4      ,9038853,
52765873)
insert into visits values (986315,     '2017-05-22',      '2017-10-14',
'2019-03-13',      'Arianne Foster', 26578.87,    7120, 2      ,81164013,
572549237)
insert into visits values (752306,     '2017-05-18',      '2017-06-07',
'2018-03-29',      'Vernon Cannon', 49014.88,    4133, 8      ,68603197,
6011962951)
insert into visits values (401071,     '2017-05-16',      '2017-10-28',
'2018-07-02',      'Cody Delacruz', 42274.98,    6434, 6      ,91761429,
362079200)
insert into visits values (189333,     '2017-01-08',      '2017-03-22',
'2019-11-14',      'Kristopher Dougherty', 91817.33 ,4630, 1      ,
55034572,      5436655387)
insert into visits values (506920,     '2017-03-23',      '2017-04-29',
'2018-05-25',      'Ahmed Wade',   66515.82,    9350, 1      ,83696889,
2985053385)
insert into visits values (930329,     '2017-03-17',      '2017-02-09',
'2018-10-14',      'Elton Johnson', 90602,      5555, 5      ,7101753,
9918294620)
insert into visits values (26217,      '2017-02-25',      '2017-05-08',
'2018-08-26',      'Sparkle Graham', 82603.56,    9355, 9      ,45730150,
929024070)
insert into visits values (54238,      '2017-02-14',      '2017-10-30',
'2018-06-25',      'Shirly Beck',   42291.77,    7426, 3      ,36328797,
4183901477)
insert into visits values (996620,     '2017-05-15',      '2017-12-15',
'2019-04-20',      'Carlo Hansen', 27828.25,    7388, 2      ,3496507,
5966772740)
insert into visits values (833698,     '2017-03-27',      '2017-10-11',
'2018-01-08',      'Beverlee Castaneda', 59993.24, 8123, 1      ,
8977331,      6626125260)
insert into visits values (4761,       '2017-04-27',      '2017-06-09',
'2019-03-03',      'Brett Randall', 61540.12,    4270, 1      ,66233683,
3492369222)
insert into visits values (773510,     '2017-05-29',      '2017-10-08',
'2019-09-03',      'Maxie Daugherty', 34098.94, 8448, 1      ,64593989
,5804237177)
insert into visits values (937667,     '2017-04-08',      '2017-11-03',
'2019-12-31',      'Madelaine Dunn', 97975.29,   4879, 1      ,93971647,

```

```

977324890)
insert into visits values (266008,      '2017-05-17',      '2017-04-16',
'2018-09-03',      'Laurene Herring', 74936.77,    4855,  5,      65321745,
229054573)
insert into visits values (799045,      '2017-01-02',      '2017-06-09',
'2018-12-09',      'Cary Duarte',     44753.40,    5726,  3,      79147483,
5864813592)
insert into visits values (388938,      '2017-05-23',      '2017-04-25',
'2019-06-08',      'Rodrigo Melton', 44289.93,    1811,  4,      95811088,
8187560116)
insert into visits values (417991,      '2017-03-23',      '2017-02-12',
'2018-12-18',      'Elinor Gay',       37963.65,    3190,  2,      95264136,
5477528289)
insert into visits values (391378,      '2017-02-14',      '2017-05-20',
'2019-06-17',      'Dodie Ford',       97281.34,    8770,  7,      83443310,
5840844404)
insert into visits values (130047,      '2017-05-09',      '2017-11-16',
'2019-02-11',      'Rigoberto Finley', 52230.82,    8901,  6,
18896384,        4063789551)
insert into visits values (840222,      '2017-02-11',      '2017-01-21',
'2018-09-17',      'Carol Vasquez',   86354.72,    5882,  2,      3548957,
2075868651)
insert into visits values (10384,       '2017-01-15',      '2017-11-05',
'2019-07-18',      'Lucius Bowers',   90846.03,    5754,  1,      20536872,
2627959571)
insert into visits values (241683,      '2017-01-09',      '2017-07-21',
'2018-05-24',      'Matthew McIntyre', 28903.56,    2128,  1,
80920878,        6975164124)
insert into visits values (501481,      '2017-04-06',      '2017-09-20',
'2018-09-09',      'Charisse Braun',  46830.73,    9051,  9,      39862515,
251747956)
insert into visits values (627588,      '2017-01-26',      '2017-10-07',
'2020-02-25',      'Kenyatta Carroll', 40056.08,    7059,  7,
50655329,        7895046015)
insert into visits values (350883,      '2017-02-02',      '2017-06-17',
'2020-02-08',      'Efren Byrd',       48206.24,    6717,  3,      4812308,
7583088441)
insert into visits values (681771,      '2017-01-16',      '2017-02-14',
'2018-11-20',      'Collette Clements', 77142.88,    9851,  2,
87915581,        6918216884)

select * from visits;

/* Creating APPOINTMENTS Table and Inserting APPOINTMENTS Data */

```

```

CREATE TABLE APPOINTMENTS (
    AppointmentID      Int          Not Null Unique,
    VisitID            Int          Not Null,
    ServiceID          Int          Not Null,
    EmployeeID         Varchar(10) Not Null,
    Primary Key (AppointmentID),
    Foreign Key (VisitID) REFERENCES VISITS (VisitID),
    Foreign Key (ServiceID) REFERENCES SERVICES (ServiceID),
    Foreign Key (EmployeeID) REFERENCES EMPLOYEES (EmployeeID))

insert into APPOINTMENTS values (769504,     8453,  1,      7583088441)
insert into APPOINTMENTS values (552000,     150336, 1,      6918216884)
insert into APPOINTMENTS values (199047,     28319,  1,      4634033099)
insert into APPOINTMENTS values (140617,     489302, 4,      7601577954)
insert into APPOINTMENTS values (293772,     986315, 2,      1303503027)
insert into APPOINTMENTS values (77719,       752306, 8,      52765873)
insert into APPOINTMENTS values (981999,     401071, 6,      572549237)
insert into APPOINTMENTS values (296873,     189333, 1,      6011962951)
insert into APPOINTMENTS values (78245,       506920, 1,      362079200)
insert into APPOINTMENTS values (112013,     930329, 5,      5436655387)
insert into APPOINTMENTS values (112384,     26217,  9,      2985053385)
insert into APPOINTMENTS values (932009,     54238,  3,      9918294620)
insert into APPOINTMENTS values (690337,     996620, 2,      5966772740)
insert into APPOINTMENTS values (867476,     833698, 1,      4183901477)
insert into APPOINTMENTS values (271937,     4761,   1,      5966772740)
insert into APPOINTMENTS values (779760,     773510, 1,      6626125260)
insert into APPOINTMENTS values (567939,     937667, 1,      3492369222)
insert into APPOINTMENTS values (664484,     266008, 5,      5804237177)
insert into APPOINTMENTS values (43058,       799045, 3,      977324890)
insert into APPOINTMENTS values (617200,     388938, 4,      229054573)
insert into APPOINTMENTS values (840011,     417991, 2,      5864813592)
insert into APPOINTMENTS values (702561,     391378, 7,      8187560116)
insert into APPOINTMENTS values (48216,      130047, 6,      )

```

```

5477528289)
insert into APPOINTMENTS values (320555,     840222,      2,
5840844404)
insert into APPOINTMENTS values (393515,     10384, 1,    4063789551)
insert into APPOINTMENTS values (390382,     241683,      1,
2075868651)
insert into APPOINTMENTS values (5159,       501481,      9,
2627959571)
insert into APPOINTMENTS values (422144,     627588,      7,
6975164124)
insert into APPOINTMENTS values (705045,     350883,      3,    251747956)
insert into APPOINTMENTS values (935619,     930329,      5
,7895046015)
insert into APPOINTMENTS values (869979,     26217, 9,    1303503027)
insert into APPOINTMENTS values (281579,     54238, 3,    52765873)
insert into APPOINTMENTS values (130592,     996620,      2,    572549237)
insert into APPOINTMENTS values (165645,     833698,      1,
6011962951)
insert into APPOINTMENTS values (977179,     4761, 1,    5436655387)
insert into APPOINTMENTS values (937753,     773510, 1,    2985053385)
insert into APPOINTMENTS values (638983,     937667,      1,
9918294620)
insert into APPOINTMENTS values (51161,      266008,      5,    229054573)
insert into APPOINTMENTS values (397679,     799045,      3,
4183901477)
insert into APPOINTMENTS values (856578,     388938,      4,
5966772740)
insert into APPOINTMENTS values (583052,     417991,      2,
6626125260)
insert into APPOINTMENTS values (62077,      391378,      7,
3492369222)
insert into APPOINTMENTS values (17992,      130047,      6,
5804237177)
insert into APPOINTMENTS values (435366,     840222,      2,    977324890)
insert into APPOINTMENTS values (586152,     773510,      1,    229054573)
insert into APPOINTMENTS values (60439,      937667,      1,
7583088441)
insert into APPOINTMENTS values (867845,     266008,      5,
6918216884)
insert into APPOINTMENTS values (985680,     799045,      3,
4634033099)
insert into APPOINTMENTS values (881980,     388938,      4,
7601577954)
insert into APPOINTMENTS values (412300,     417991,      2,
1303503027)

```

```

insert into APPOINTMENTS values (445774,      391378,      7,      52765873)
insert into APPOINTMENTS values (922622,      130047,      6,      572549237)
insert into APPOINTMENTS values (191304,      840222,      2,
6011962951)
insert into APPOINTMENTS values (131115,      10384, 1,      362079200)
insert into APPOINTMENTS values (982579,      241683,      1,
5436655387)
insert into APPOINTMENTS values (467918,      501481,      9,
2985053385)
insert into APPOINTMENTS values (429189,      627588,      7,
9918294620)
insert into APPOINTMENTS values (707491,      350883 ,3,      7583088441)
insert into APPOINTMENTS values (802776,      681771,      2,
4183901477)
insert into APPOINTMENTS values (760380,      986315,      2,
5966772740)

select * from APPOINTMENTS;

/* Creating CORPORATE Table and Inserting CORPORATE Data */

CREATE TABLE CORPORATE(
    ContactName Char(255)  Not Null,
    ContactPhone varchar(10)      Not Null,
    ClientID integer not null unique,
    Foreign Key (ClientID) REFERENCES CLIENTS (ClientID))

insert into CORPORATE values ('Raddie Velareal', 7868105160, 9038853)
insert into CORPORATE values ('Reggie Renton', 9155973136, 81164013)
insert into CORPORATE values ('Lynelle Frowde', 8624397881, 68603197)
insert into CORPORATE values ('Dave Linthead', 8506516447, 91761429)
insert into CORPORATE values ('Roger Canet', 7045556346, 55034572)
insert into CORPORATE values ('Bev Burras', 9187985896, 83696889)
insert into CORPORATE values ('Katalin Ladbury', 4083015743, 7101753)
insert into CORPORATE values ('Bronnie Cherry', 4045406276, 45730150)
insert into CORPORATE values ('Betteann Braunton', 8634429086,
36328797)
insert into CORPORATE values ('Hobie Bumby' ,5638956957 ,3496507)
insert into CORPORATE values ('Alfonse Ebhardt', 5858746191, 8977331)
insert into CORPORATE values ('Linet Terris', 6081403315, 20536872)
insert into CORPORATE values ('Jillene Penna', 3145806106, 80920878)
insert into CORPORATE values ('Sharyl Widdop', 6153672182, 39862515)
insert into CORPORATE values ('Jamil Borkett', 4132574932, 50655329)

```

```

select * from CORPORATE;

/* Creating INDIVIDUAL Table and Inserting INDIVIDUAL Data */

CREATE TABLE INDIVIDUAL(
    CreditCard  varchar(16),
    ClientID integer not null unique,
    Foreign Key (ClientID) REFERENCES CLIENTS (ClientID)

    insert into INDIVIDUAL values (392465468562285,      47620087)
    insert into INDIVIDUAL values (737875881691977,      61406868)
    insert into INDIVIDUAL values (915428977813510,      8565248)
    insert into INDIVIDUAL values (205430679934836,      66233683)
    insert into INDIVIDUAL values (452468253896930,      64593989)
    insert into INDIVIDUAL values (995621911597674,      93971647)
    insert into INDIVIDUAL values (787617247129312,      65321745)
    insert into INDIVIDUAL values (794980363179730,      79147483)
    insert into INDIVIDUAL values (466604156551278,      95811088)
    insert into INDIVIDUAL values (224623828193391,      95264136)
    insert into INDIVIDUAL values (431158481165622,      83443310)
    insert into INDIVIDUAL values (416814892773282,      18896384)
    insert into INDIVIDUAL values (523083297614496,      3548957)
    insert into INDIVIDUAL values (350101724221769,      4812308)
    insert into INDIVIDUAL values (933813096207303,      87915581)

select * from INDIVIDUAL;

/* Updating the PROPERTIES table to have more realistic
values for sqft/bed/bath that aren't randomized */

BEGIN TRANSACTION

    update properties set SqFt = 1935 where PropertyID = 9350;
    update properties set #Bed = 3 where PropertyID = 9350;
    update properties set #Bath = 2 where PropertyID = 9350;

    update properties set SqFt = 3500 where PropertyID = 4855;
    update properties set #Bed = 4 where PropertyID = 4855;
    update properties set #Bath = 3 where PropertyID = 4855;

    update properties set SqFt = 4520 where PropertyID = 5882;
    update properties set #Bed = 6 where PropertyID = 5882;

```

```
update properties set #Bath = 4 where PropertyID = 5882;

update properties set SqFt = 525 where PropertyID = 6717;
update properties set #Bed = 1 where PropertyID = 6717;
update properties set #Bath = 1 where PropertyID = 6717;

update properties set SqFt = 2200 where PropertyID = 5726;
update properties set #Bed = 3 where PropertyID = 5726;
update properties set #Bath = 2 where PropertyID = 5726;

update properties set SqFt = 1080 where PropertyID = 5555;
update properties set #Bed = 2 where PropertyID = 5555;
update properties set #Bath = 1 where PropertyID = 5555;

update properties set SqFt = 2200 where PropertyID = 1811;
update properties set #Bed = 3 where PropertyID = 1811;
update properties set #Bath = 2 where PropertyID = 1811;

update properties set SqFt = 2200 where PropertyID = 8770;
update properties set #Bed = 3 where PropertyID = 8770;
update properties set #Bath = 2 where PropertyID = 8770;

update properties set SqFt = 2985 where PropertyID = 5754;
update properties set #Bed = 4 where PropertyID = 5754;
update properties set #Bath = 3 where PropertyID = 5754;

update properties set SqFt = 1500 where PropertyID = 7388;
update properties set #Bed = 2 where PropertyID = 7388;
update properties set #Bath = 2 where PropertyID = 7388;

update properties set SqFt = 6250 where PropertyID = 4270;
update properties set #Bed = 8 where PropertyID = 4270;
update properties set #Bath = 5 where PropertyID = 4270;

update properties set SqFt = 2985 where PropertyID = 8123;
update properties set #Bed = 4 where PropertyID = 8123;
update properties set #Bath = 3 where PropertyID = 8123;

update properties set SqFt = 1080 where PropertyID = 1106;
update properties set #Bed = 2 where PropertyID = 1106;
update properties set #Bath = 1 where PropertyID = 1106;

update properties set SqFt = 1935 where PropertyID = 4630;
update properties set #Bed = 3 where PropertyID = 4630;
update properties set #Bath = 2 where PropertyID = 4630;
```

```

update properties set SqFt = 2670 where PropertyID = 1001;
update properties set #Bed = 3 where PropertyID = 1001;
update properties set #Bath = 3 where PropertyID = 1001;

update properties set SqFt = 2985 where PropertyID = 7426;
update properties set #Bed = 4 where PropertyID = 7426;
update properties set #Bath = 3 where PropertyID = 7426;

update properties set SqFt = 2670 where PropertyID = 9051;
update properties set #Bed = 3 where PropertyID = 9051;
update properties set #Bath = 3 where PropertyID = 9051;

update properties set SqFt = 1935 where PropertyID = 6929;
update properties set #Bed = 3 where PropertyID = 6929;
update properties set #Bath = 2 where PropertyID = 6929;

update properties set SqFt = 2670 where PropertyID = 3190;
update properties set #Bed = 3 where PropertyID = 3190;
update properties set #Bath = 3 where PropertyID = 3190;

update properties set SqFt = 1080 where PropertyID = 9851;
update properties set #Bed = 2 where PropertyID = 9851;
update properties set #Bath = 1 where PropertyID = 9851;

update properties set SqFt = 1050 where PropertyID = 6789;
update properties set #Bed = 2 where PropertyID = 6789;
update properties set #Bath = 1 where PropertyID = 6789;

update properties set SqFt = 2985 where PropertyID = 4233;
update properties set #Bed = 4 where PropertyID = 4233;
update properties set #Bath = 3 where PropertyID = 4233;

update properties set SqFt = 1935 where PropertyID = 8901;
update properties set #Bed = 3 where PropertyID = 8901;
update properties set #Bath = 2 where PropertyID = 8901;

update properties set SqFt = 2980 where PropertyID = 8448;
update properties set #Bed = 4 where PropertyID = 8448;
update properties set #Bath = 3 where PropertyID = 8448;

update properties set SqFt = 2985 where PropertyID = 7059;
update properties set #Bed = 4 where PropertyID = 7059;
update properties set #Bath = 3 where PropertyID = 7059;

```

```

update properties set SqFt = 2670 where PropertyID = 7120;
update properties set #Bed = 3 where PropertyID = 7120;
update properties set #Bath = 3 where PropertyID = 7120;

update properties set SqFt = 1935 where PropertyID = 9355;
update properties set #Bed = 3 where PropertyID = 9355;
update properties set #Bath = 2 where PropertyID = 9355;

update properties set SqFt = 2670 where PropertyID = 3016;
update properties set #Bed = 3 where PropertyID = 3016;
update properties set #Bath = 3 where PropertyID = 3016;

update properties set SqFt = 1500 where PropertyID = 2128;
update properties set #Bed = 2 where PropertyID = 2128;
update properties set #Bath = 2 where PropertyID = 2128;

update properties set SqFt = 6250 where PropertyID = 4879;
update properties set #Bed = 8 where PropertyID = 4879;
update properties set #Bath = 5 where PropertyID = 4879;

update properties set SqFt = 2985 where PropertyID = 8000;
update properties set #Bed = 4 where PropertyID = 8000;
update properties set #Bath = 3 where PropertyID = 8000;

update properties set SqFt = 1080 where PropertyID = 6434;
update properties set #Bed = 2 where PropertyID = 6434;
update properties set #Bath = 1 where PropertyID = 6434;

update properties set SqFt = 2200 where PropertyID = 4133;
update properties set #Bed = 3 where PropertyID = 4133;
update properties set #Bath = 2 where PropertyID = 4133;

update properties set SqFt = 1100 where PropertyID = 1000;
select * from properties order by SqFt desc;

rollback; -- only perform if incorrect
commit; -- only perform if correct and ready to change the database

/* QUERIES TO TEST INTEGRITY RULES */

/* Employee and Owner Names with Date of Visit and Total Cost */
    select EFName , ELName , TotalCost , Owner , DateOfVisit
    from employees , visits
    where employees.employeeid = visits.employeeid

```


SQL Scripts of Client Request Queries

```
/* State that has the most owners. */

select distinct top 1
    ClState as 'State with most owners',
    count(ClState) as 'Number of owners'
from clients
    group by ClState
    order by count(ClState) desc;

/* 2. Number/list of owners in each state. */

select      ClientID, ClState as 'State'
                CFName as 'First Name', CLName as 'Last Name'
from CLIENTS
order by ClState, CFName asc;

select distinct
    ClState as 'State',
    count(ClState) as 'Number of owners'
from CLIENTS
    group by ClState
    order by count(ClState) desc;

/* 3. Total number of visits for each client for each season. */

Select CFName, CLName, Count(VisitID) as 'Number of Visits',
    (case when month(DateOfVisit) in (12, 1, 2) then 'Winter'
     when month(DateOfVisit) in 3, 4, 5 then 'Spring'
     when month(DateOfVisit) in 6, 7, 8 then 'Summer'
     when month(DateOfVisit) in 9, 10, 11 then 'Autumn'
     end) as 'Season'
    From Clients C
    JOIN VISITS V
        on C.ClientID = V.ClientID
        Group By CFName, CLName, DateOfVisit;

/* 4. Total charge for each visit where there are multiple services
performed. */
```

```
select Count(VisitID)as 'Number of Visits', Owner, Sum(TotalCost) as 'Total Cost'  
    From VISITS  
        Group by VisitID, Owner  
            Having Count(VisitID) > 2;
```

```
/* 5. A list of clients who have ordered pest control. */
```

```
Select C.ClientID, CFName, CLName, ServiceDesc  
    From Clients C  
    JOIN VISITS V  
        on C.ClientID = V.ClientID  
    JOIN SERVICES S  
        on V.ServiceID = S.ServiceID  
    Where ServiceDesc = 'Pest Control';
```

```
/* 6. The service taking the most time/number of employees. */
```

```
Select ServiceDesc,  
    max(datediff(day, startdate, enddate)) as 'Number of Days',  
    count(e.employeeid) as 'Number of Employees'  
    From SERVICES S  
    JOIN VISITS V  
        on S.ServiceID = V.ServiceID  
    JOIN EMPLOYEES E  
        on E.EmployeeID = V.EmployeeID  
        Group By ServiceDesc;
```

```
/* 7. Employees residing in the same state as a client. */
```

```
Select Owner, EFName, ELName, C.ClState, E.State  
    From VISITS V  
    JOIN EMPLOYEES E  
        on V.EmployeeID = E.EmployeeID  
    JOIN CLIENTS C  
        on C.ClientID = V.ClientID  
        Where C.ClState = E.State
```

```
/* 8. List of employees performing the most services. */
```

```

SELECT A.EmployeeID, EFName, ELName, COUNT(ServiceID) AS 'Number of
Services'
    FROM EMPLOYEES E
    JOIN APPOINTMENTS A ON E.EmployeeID = A.EmployeeID
        GROUP BY A.EmployeeID, EFName, ELName
            ORDER BY COUNT(ServiceID) DESC;

/* 9. Total revenue received last year. */
SELECT SUM(TotalCost) AS 'Total Revenue for 2019'
    FROM VISITS
        WHERE DateOfVisit LIKE '2019%';

/* 10. Total damaged caused by the last hurricane. */
SELECT SUM(TotalCost) AS 'Total Cost from Most Recent Hurricane ($)'
    FROM VISITS
        WHERE ServiceID IN (
            SELECT ServiceID
                FROM SERVICES
                    WHERE ServiceDesc = 'Hurricane Recovery');

/* 11. Employees making more than the average rate for their job
specification. */

SELECT EmployeeID, EFName, ELName, HourlyRate, SpecializationID
    FROM EMPLOYEES
        WHERE SpecializationID = 1
            AND HourlyRate > (
                SELECT AVG(HourlyRate)
                    FROM EMPLOYEES
                        WHERE SpecializationID = 1);

SELECT EmployeeID, EFName, ELName, HourlyRate, SpecializationID
    FROM EMPLOYEES
        WHERE SpecializationID = 1
            AND HourlyRate > (
                SELECT AVG(HourlyRate)
                    FROM EMPLOYEES
                        WHERE SpecializationID = 2);

SELECT EmployeeID, EFName, ELName, HourlyRate, SpecializationID
    FROM EMPLOYEES
        WHERE SpecializationID = 1

```

```

        AND HourlyRate > (
            SELECT AVG(HourlyRate)
            FROM EMPLOYEES
            WHERE SpecializationID = 3);

SELECT EmployeeID, EFName, ELName, HourlyRate, SpecializationID
FROM EMPLOYEES
WHERE SpecializationID = 1
AND HourlyRate > (
    SELECT AVG(HourlyRate)
    FROM EMPLOYEES
    WHERE SpecializationID = 4);

SELECT EmployeeID, EFName, ELName, HourlyRate, SpecializationID
FROM EMPLOYEES
WHERE SpecializationID = 1
AND HourlyRate > (
    SELECT AVG(HourlyRate)
    FROM EMPLOYEES
    WHERE SpecializationID = 5);

SELECT EmployeeID, EFName, ELName, HourlyRate, SpecializationID
FROM EMPLOYEES
WHERE SpecializationID = 1
AND HourlyRate > (
    SELECT AVG(HourlyRate)
    FROM EMPLOYEES
    WHERE SpecializationID = 6);

SELECT EmployeeID, EFName, ELName, HourlyRate, SpecializationID
FROM EMPLOYEES
WHERE SpecializationID = 1
AND HourlyRate > (
    SELECT AVG(HourlyRate)
    FROM EMPLOYEES
    WHERE SpecializationID = 7);

SELECT EmployeeID, EFName, ELName, HourlyRate, SpecializationID
FROM EMPLOYEES
WHERE SpecializationID = 1
AND HourlyRate > (
    SELECT AVG(HourlyRate)
    FROM EMPLOYEES
    WHERE SpecializationID = 8);

```

```

        WHERE SpecializationID = 8);

SELECT EmployeeID, EFName, ELName, HourlyRate, SpecializationID
  FROM EMPLOYEES
    WHERE SpecializationID = 1
      AND HourlyRate > (
        SELECT AVG(HourlyRate)
          FROM EMPLOYEES
            WHERE SpecializationID = 9);

SELECT EmployeeID, EFName, ELName, HourlyRate, SpecializationID
  FROM EMPLOYEES
    WHERE SpecializationID = 1
      AND HourlyRate > (
        SELECT AVG(HourlyRate)
          FROM EMPLOYEES
            WHERE SpecializationID = 10);

SELECT EmployeeID, EFName, ELName, HourlyRate, SpecializationID
  FROM EMPLOYEES
    WHERE SpecializationID = 1
      AND HourlyRate > (
        SELECT AVG(HourlyRate)
          FROM EMPLOYEES
            WHERE SpecializationID = 11);

/* 12. Clients requiring multiple visits for a single service. */
SELECT COUNT(VisitID) AS 'Visits for Same Service', ClientID, ServiceID
  FROM VISITS
 GROUP BY ServiceID, ClientID
   HAVING COUNT(VisitID) > 1
 ORDER BY ClientID;

```

SQL Scripts of Basic Tests

```
/*
-----
Basic Tests
-----
*/



/*      Basic Test #1
Adding a new client with no last name (e.g. a corporation)
and a Canadian zip code (6 letters and numbers). This query should
succeed.
*/
SELECT *
    FROM CLIENTS
        WHERE CLName IS NULL
        AND CLZip LIKE 'a%';

INSERT INTO CLIENTS
    VALUES (2738192, '123 South 5th St', 1827762154, 2, 'djskdioe@gmail.com',
'AZ', 'Tempe', 'a12b12', 'Walmart', NULL);





/*      Basic Test #2
Adding a new client without a phone number. This query should NOT
succeed. */
SELECT *
    FROM CLIENTS
        WHERE ClPhone IS NULL;

INSERT INTO CLIENTS
    VALUES (2781928, '25 E 17th Ave', NULL, 2, 'dbgv2dioe@gmail.com', 'CA',
'Orange', '126716', 'Michael', 'David');





/*      Basic Test #3
Deleting a visit and all visit details should succeed using one
statement.
*/
begin transaction
    delete from VISITS
```

```

where VisitID = 681771;
--rollback;

select * from VISITS;

/*
   Basic Test #4
   Two INSERT statements executed together - the first one adding a new
Client, the second one
      adding a new Property for that Client, should succeed. */

select * from clients;
select * from properties;

insert into clients values (5000000, '123 Main St' , 5555555555, 1,
'test@test.com', 'AZ', 'Phoenix', 'Test','Subject', 10000)
insert into properties values (1000, 'Subject', '123 Main St', 1, 1 , 1111.11,
2019, 'tile', 'brick', 5000000)

select * from clients;
select * from PROPERTIES;

/*
   Basic Test #5
   Two similar INSERT statements (#4) executed together in the opposite
order should fail. */

select * from CLIENTS;
select * from PROPERTIES;

insert into properties values (1234, 'Fail', '111 West Ave', 1, 2, 12345.43,
1929, 'concrete', 'vinyl', 1234567)
insert into clients values (1234567, '111 West Ave', 1234567890, 19,
'pleasefail@test.com', 'TX', 'Austin', 'Please', 'Fail',12345)

select * from properties;
select * from clients;

/*
   Basic Test #6
   Adding a new employee to complete an existing service should succeed. */

```

```

begin transaction
update APPOINTMENTS
    set EmployeeID = 5804237177
        -- set to employee ID 5804237177
        -- was initially 2627959571
    where AppointmentID = 5159

select * from APPOINTMENTS
    where AppointmentID = 5159;

--rollback;

/*
    Basic Query #7
    Construct 4 other preliminary tests to show that your database design
works.
    Select tests that you know will make your design fail where it should
fail.
    (Adding a new client without a name should fail). */

insert into clients
values (34234324, null, '15 Walton Pass', '8608999058', 16,
    'hrenton0@ted.com', 'CT', 'Hartford')

/*
    Basic Query #8
    Construct 4 other preliminary tests to show that your database design
works.
    Select tests that you know will make your design fail where it should
fail.
    (This will fail as it is repeating employee Id.)*/

INSERT INTO EMPLOYEES
    (EmployeeId, EFName, ELName, HourlyRate, EAddress, EPhone,
SpecializationID, State, City)
Values (4063789551, 'Sasha', 'ABC', 50.01, '123 Main St', '1234667989', 1,
'NY', 'New York City' )

/*
    Basic Query #9
    Construct 4 other preliminary tests to show that your database design
works.
    Select tests that you know will make your design fail where it should
fail.
    (This will fail as it is inserting big specialized id value which cant be
*/

```

```
held by tinyint)*/  
  
INSERT INTO EMPLOYEES  
    (EmployeeId, EFName, E1Name, HourlyRate, EAddress, EPPhone,  
SpecializationID, State, City)  
Values (40637895667, 'AnyName', 'ABC', 50, '123 Main St', '1234667989', 231231,  
'NY', 'New York City' )  
  
/*      Basic Query #10  
        Construct 4 other preliminary tests to show that your database design  
works.  
        Select tests that you know will make your design fail where it should  
fail.  
        (It will fail because it is supposed to be 6 digits max but we have  
inserted more than the required value). */  
  
Insert into Services Values(1, 'Cleaning', 1123123222.99)
```

SQL Scripts of Transactions

```
/* 1. (5) When adding a new client, also update the corresponding 'child' table. Add the property for the client and link the client with the property. */

*/declare @clientId INT,@propertyId INT

SELECT @clientId = MAX(ClientID) + 1 FROM CLIENTS;
SELECT @propertyId = MAX([PropertyID]) + 1 FROM PROPERTIES;

SELECT * FROM CLIENTS WHERE CLIENTID=@clientId
SELECT * FROM PROPERTIES WHERE PropertyID=@propertyId

BEGIN TRANSACTION _Test

INSERT INTO
CLIENTS([ClientID],[C1Name],[C1Address],[C1Phone],[YearsServiced],[C1Email]
,[C1State],[C1City])
VALUES(@clientId, 'Testing', '12 George Street',
'1234567891',20,'testing@yahoo.com','AZ','PHOENIX')
INSERT INTO
PROPERTIES([PropertyID],[PropName],[Address],[#Bed],[#Bath],[SqFt],[YrBuilt]
,[RoofType],[ExteriorType],[ClientID])
VALUES(@propertyId, 'xyz', '12 Square Road', 5, 3,
3456.35,2008,'tile','brick',@clientId)

ROLLBACK TRANSACTION _Test
--COMMIT TRANSACTION _Test

SELECT * FROM CLIENTS WHERE CLIENTID=@clientId
SELECT * FROM PROPERTIES WHERE PropertyID=@propertyId
```

```
/* 2. (5) When adding a visit request, also include what is requested in  
the service request table. */  
  
BEGIN TRANSACTION;  
  
Insert Into VISITS Values (999999, '2017-06-05', '2017-07-02', '2017-08-  
24', 'Taylor Swift', 100.00, 1001, 1, 1000000, 1000000000);  
Insert Into APPOINTMENTS Values (424242, 681771, 10, 4063789551);  
  
Select * From VISITS Where VisitID = 999999  
Select * From APPOINTMENTS Where AppointmentID = 424242;  
  
Rollback;  
Commit;
```

SQL Scripts of Advanced Tests, Optimizations, & Views

```
/*
-----
Advanced Tests
-----
*/



/*      Advanced Query #1:
    Find the number of clients per state for every state with more than 1
client.
    Also lists the average years serviced per state.
*/
select
    ClState as 'State',
    count(ClState) as 'Number of Clients',
    avg(YearsServiced) as 'Average Years Serviced'
from CLIENTS
    group by ClState
    having count(ClState) > 1; --states with more than 1 client


/*
      Advanced Query #2:
    Lists the number of services performed by service type,
    where there have been 5 or more services performed
*/
select
    S.ServiceDesc as 'Service Type',
    count(A.ServiceID) as '# of Services Performed'
from APPOINTMENTS A
    join SERVICES S
        on A.ServiceID = S.ServiceID
    group by S.ServiceDesc
    having count(A.ServiceID) >= 5;


/*
      Advanced Query #3
    Lists the total combined salaries for each specialization,
    broken down by state, where the total combined salary is less than $75.
*/
select
```

```

        sum(HourlyRate) as 'Total Hourly Salary',
        SpecializationID as 'Specialization',
        State
    from EMPLOYEES
    group by SpecializationID, State
    having sum(HourlyRate) < 75;

/*
   Advanced Query #4
   Fully NESTED query (no joins) that combines 3 or more data tables */

SELECT EPhone
    FROM EMPLOYEES
    WHERE EmployeeID IN (
        SELECT EmployeeID
            FROM APPOINTMENTS
            WHERE ServiceID IN (
                SELECT ServiceID
                    FROM SERVICES
                    WHERE ServiceRate = 60.00));
5.
/*
   Advanced Query #5
   Fully NESTED query (no joins) that combines 3 or more data tables */

SELECT PropName
    FROM PROPERTIES
    WHERE PropertyID IN (
        SELECT PropertyID
            FROM VISITS
            WHERE ClientID IN (
                SELECT ClientID
                    FROM CLIENTS
                    WHERE ClPhone = 7042993362));
6.
/*
   Advanced Query #6
   Fully NESTED query (no joins) that combines 3 or more data tables */

SELECT *
    FROM SERVICES
    WHERE ServiceID IN (
        SELECT ServiceID
            FROM APPOINTMENTS
            WHERE VisitID IN (

```

```

        SELECT VisitID
        FROM VISITS
        WHERE ClientID IN (
            SELECT ClientID
            FROM INDIVIDUAL
            WHERE
CreditCard = 466604156551278));

/*
    Advanced Query #7
    3 must be Views (Store results that will be of interest to your client)
*/

Create View VeteranCustomers as
Select PropertyID, PropName, ClAddress, YearsServiced
    From Clients, Properties
        Where Clients.ClientID = Properties.ClientID
        AND YearsServiced > 5;

Select *
    From VeteranCustomers
        Order by YearsServiced desc;

/*
    Advanced Query #8
    3 must be Views (Store results that will be of interest to your client)
*/
Create View BusyEmployees as
Select E.EmployeeID, EFName , ELName , Count(AppointmentID) as 'Number of
Appointments'
    From Employees as E, APPOINTMENTS as A
        Where E.EmployeeID = A.EmployeeID
        Group by E.EmployeeID, EFName, ELName

Select * from BusyEmployees
    Order By [Number of Appointments] desc;

/*
    Advanced Query #9
    3 must be Views (store results that will be of interest to your client)
*/
/*
    Finding the total cost of modern homes (Built in 2000 or newer) that are
in the same state as Rodrigo Melton */

Create View ModernHomeServiceCost as

```

```
Select Owner , YrBuilt as 'Year Built' , ClCity as 'City' , ClState as 'State'
, TotalCost
    From VISITS, PROPERTIES, CLIENTS
        Where CLIENTS.ClientID = VISITS.ClientID
        AND PROPERTIES.PropertyID = VISITS.PropertyID
        AND ClState in (Select ClState from CLIENTS where CFName =
'Rodrigo' AND CLName = 'Melton')
                AND YrBuilt > 1999;

Select * From ModernHomeServiceCost
    Order by TotalCost desc;
```

SQL Scripts of Trigger

```
/* Trigger */
/* 1. (10) TRIGGER: Process a service (visit) request. When a new request is
added,
the trigger will automatically assign an employee who has the required
specialization
(if the work is plumbing, assign the corresponding employee with a plumber
specialization). */

create trigger Appointment_Specialization
    on APPOINTMENTS
        after insert
as
    update APPOINTMENTS
        set EmployeeID =
            ((select EmployeeID
                from EMPLOYEES
                where SpecializationID in
                    (select ServiceID
                        from INSERTED)))
    from APPOINTMENTS
    join INSERTED I on I.EmployeeID = APPOINTMENTS.EmployeeID
    where APPOINTMENTS.AppointmentID = I.AppointmentID;

begin transaction
insert into APPOINTMENTS values (42424242, 681771, 10, 1303503027);

rollback;      --execute if data is incorrect
commit;          --only execute if data is correct and ready to be
implemented

select * from APPOINTMENTS where AppointmentID = 42424242;
select * from employees where SpecializationID = 10;
```

SQL Scripts of Procedure

```
/* Procedure */
/* 2. (10) PROCEDURE: When an employee has been assigned to perform service
and completes that
service, add a new record to the ServiceCompletionLog to record the season,
start date, number of
days to complete the service, property ID, employee name, specialization.
```

Hint: You will need to create a ServiceCompletionLog to keep track of this data. The procedure will be executed after the service has been completed (add the end date to the record). */

```
CREATE TABLE SERVICECOMPLETIONLOG(
    VisitID          int,
    StartDate        date,
    EndDate          date,
    Season           char(6),
    DaysToFin        varchar(4),
    PropertyID       int,
    EFName           char(30),
    ELName           char(30),
    Specialization   tinyint,
    Foreign Key (VisitID) REFERENCES VISITS (VisitID),
    Foreign Key (PropertyID) REFERENCES PROPERTIES (PropertyID));
```

```
CREATE PROCEDURE ServiceComp
    @VisitID int
    AS
    BEGIN
        INSERT INTO SERVICECOMPLETIONLOG (VisitID, StartDate, EndDate,
Season, DaysToFin, PropertyID,
        EFName, ELName, Specialization)
        VALUES (@VisitID,
        (SELECT StartDate
        FROM VISITS
            WHERE @VisitID = VisitID),
        (SELECT EndDate
        FROM VISITS
            WHERE @VisitID = VisitID),
        (CASE
        WHEN (SELECT EndDate FROM VISITS WHERE @VisitID =
```

```

VisitID) LIKE '%-03-%'
          OR
          (SELECT EndDate FROM VISITS WHERE @VisitID = VisitID)
LIKE '%-04-%'
          OR
          (SELECT EndDate FROM VISITS WHERE @VisitID = VisitID)
LIKE '%-05-%'
          THEN 'Spring'
          WHEN (SELECT EndDate FROM VISITS WHERE @VisitID =
VisitID) LIKE '%-06-%'
          OR
          (SELECT EndDate FROM VISITS WHERE @VisitID = VisitID)
LIKE '%-07-%'
          OR
          (SELECT EndDate FROM VISITS WHERE @VisitID = VisitID)
LIKE '%-08-%'
          THEN 'Summer'
          WHEN (SELECT EndDate FROM VISITS WHERE @VisitID =
VisitID) LIKE '%-09-%'
          OR
          (SELECT EndDate FROM VISITS WHERE @VisitID = VisitID)
LIKE '%-10-%'
          OR
          (SELECT EndDate FROM VISITS WHERE @VisitID = VisitID)
LIKE '%-11-%'
          THEN 'Fall'
          ELSE 'Winter'
          END),
DATEDIFF(day,
          (SELECT StartDate
           FROM VISITS
           WHERE @VisitID = VisitID),
          (SELECT EndDate
           FROM VISITS
           WHERE @VisitID = VisitID)),
          (SELECT PropertyID
           FROM VISITS
           WHERE @VisitID = VisitID),
          (SELECT EFName
           FROM Employees
           WHERE EmployeeID IN (
           SELECT EmployeeID
           FROM VISITS
           WHERE @VisitID = VisitID))

```

```
        WHERE VisitID =
@VisitID)),  
        (SELECT ELName  
         FROM Employees  
         WHERE EmployeeID IN (  
             SELECT EmployeeID  
              FROM VISITS  
              WHERE VisitID =  
@VisitID)),  
        (SELECT SpecializationID  
         FROM Employees  
         WHERE EmployeeID IN (  
             SELECT EmployeeID  
              FROM VISITS  
              WHERE VisitID =  
@VisitID))));  
    END  
  
SELECT *  
  FROM SERVICECOMPLETIONLOG;  
  
EXEC ServiceComp @VisitID = 26217;
```

Index/Query Plan & Optimizations

```
/* Index/Query Plan 1 */
/* 1. (2) Identify the attributes (at least 2) that should have secondary
indexes defined (cluster index is
already created) to speed up the query you will define in #3. Take a snapshot
of the tables you will use
(select * from each table and show the attributes and a few rows of data in
each one). */

SELECT *
    FROM PROPERTIES;

SELECT *
    FROM VISITS;

/* Index/Query Plan 2 */
/* 2. (2) Write the SQL commands to create the secondary indexes for the 2
attributes identified in #1. Give
reasons for each attribute selected (use comments). Take a snapshot of the SQL
code for both indexes. */

CREATE INDEX Index#Bed ON PROPERTIES (#Bed);
/* Attribute selected because we would like to quickly search properties by how
many bedrooms they have
because it is one of the most important property attributes for both clients
and the company. */

CREATE INDEX IndexVisitDate ON VISITS(DateOfVisit);
/* Attribute selected because we would like to quickly search visits by which
date they occurred on
so we can then investigate any issues that may have occurred during a visit on
a certain date. */

/* Index/Query Plan 3 */
/* 3. (2) Create an advanced join query and execute (use any multiple tables
you wish). Show the query, the
data results, the time to complete, and the query plan (take snapshots of
each). The idea here is to
create a query that will tax the computer resources - force the DBMS to use its
resources. */

SET STATISTICS TIME ON;

SELECT CFName, ClPhone, ClCity
```

```

FROM CLIENTS C
JOIN VISITS V ON V.ClientID = C.ClientID
JOIN PROPERTIES P ON P.PropertyID = V.PropertyID
JOIN SERVICECOMPLETION S ON V.VisitID = S.VisitID
    WHERE CFName IN
('Sparkle','Modesto','Kenyatta','Kristopher','Fredricka','Maxie')
    AND #Bed > 3
    AND S.PropertyID > #Bath;

/* Index/Query Plan 4 (including optimization) */
/* 4. (2) Optimize & execute the query in #3. Show the query, the data results,
the time to complete, and
the query plan (take snapshots of each). */

SET STATISTICS TIME ON;

SELECT CFName, ClPhone, ClCity
    FROM CLIENTS C
        WHERE ClientID IN (
            SELECT ClientID
                FROM VISITS          -- JOINING VISITS TABLE BY
CONNECTING V.ClientID = C.ClientID
                WHERE CFName IN
('Sparkle','Modesto','Kenyatta','Kristopher','Fredricka','Maxie') -- RESTRAINT
FROM ORIGINAL QUERY
                AND PropertyID IN (
                    SELECT PropertyID
                        FROM PROPERTIES          --
JOINING PROPERTIES TABLE BY CONNECTING P.PropertyID = V.PropertyID
                        WHERE #Bed > 3          --
RESTRAINT FROM ORIGINAL QUERY
                AND PropertyID IN (
                    SELECT PropertyID
                        FROM
SERVICECOMPLETION          -- JOINING SERVICECOMPLETION TABLE VIA V.VisitID =
S.VisitID
                        WHERE
PropertyID > #Bath));      -- RESTRAINT FROM ORIGINAL QUERY

```