FALL 2021 – SFWR ENG 3DB3

Assignment 2 Part III: Indexes

Submitted by: Hamrish Saravanakumar (saravah)

CREATE INDEX OrderDOBIND ON Order(DateOfBirth);

- This index would be defined only on the DataOfBirth attribute of the Order table.
- Since the purpose of this index would be to be able to effectively query a range of ages, a B+ Tree Index would be used. The order of index data entries is not the same as the order of data records, meaning that this would be an unclustered index.
- This index would also be effective for question 2 to query the dateOfBirths for order placers such that their ages are between 20 and 35 (Order.DateOfBirth <= '2001-11-05' AND Order.DateOfBirth >= '1986-11-05')
- Additionally, this index would also be effective for question one to specifically query through the range of dates of birth such that the person is older than the age of 18 (Order.DateOfBirth <= '2003-11-05')

CREATE INDEX OrderDOBIND ON Order(DateOfBirth, Date);

- This index would be defined on both the DateOfBirth and Date attributes of the Order table
- The purpose of this index would be both to effectively query a range of dates on either the dateOfBirth or Date attributes, but also to account for equality indexing on specific dates. For this reason, a composite B+ Tree Index should be used. The order of index data entries is not the same as the order of data records, meaning that this would be an unclustered index.
- This index would perform well as a composite index, especially for question one, where the attribute Order.Date would also be used to perform a range query for individuals older than 18 years of age (Order.DateOfBirth <= '2003-11-05') and those who ordered on July 22nd 2020 (Order.Date = '2020-07-22').

CREATE INDEX PersonPostalCodeIND ON Person(PostalCode);

- This index would be defined only on the PostalCode attribute of the Person table.
- The purpose of this index would be to effectively obtain results of people who have a particular postal code. Being an equality query, a hash index is the most suitable type of index. The order of index data entries is not the same as the order of data records, meaning that this would be an unclustered index.
- Specifically, this index would aid the query for question 4b in order to quickly return the postal codes that start with the letter ‘M’ by indexing on Person.PostalCode. This would replace the operation (WHERE LEFT(Person.PostalCode, 1) = 'M')

CREATE INDEX WriteReviewIND ON WriteReview(FirstName);

- This index would be defined only on the FirstName attribute of the WriteReview table
- The purpose of this index would be to effectively index for the maximum count value of the FirstName attribute. This specifically would be used for question 3 in order to determine the name of the customer(s) who has/have the most reviews written.
- In order to implement this index, a B+ tree index should be used to find the maximum value by immediately scanning the ordered index in descending order. The order of the data records and index data entries should be the same in order for this ordered index to cocur, making the index clustered.