



Architektur und Implementierung von Datenbanksystemen

Meilenstein 4

Team 3 - Gründlinger Diana, Huber Marcel, Klotz Thomas, Targa Aaron, Thalmann Matthias

Anforderungen

Hash Index:

- Tabelle mit allen Indizes erstellen
- Index Seek Implementation
- Anwendung bei Optimierung

Handhabung von Randbedingungen bei Selection Pushdown

Index Erstellung

- Anlegen von Python Dictionaries für alle Index Spalten
- Der Index wird dann beim hinzufügen der Daten der Tabelle mit den hinzugehörigen Reihen gefüllt

```
1 def _create_indices(table_name, schema: Schema, index_start, index_lines):
2     _indices[table_name] = dict()
3     for i, line in enumerate(index_lines):
4         line = line.rstrip('\n')
5
6         try:
7             schema.get_column_index(line)
8         except TableIndexException:
9             raise TableParsingException(
10                 f'Column "{line}" in line {i + 2 + index_start} does not exist in schema')
11
12     _indices[table_name][line] = dict()
```

Index Tabelle

- Holt alle Indizes vom internen Index Dictionary
- Erstellt dann Tabelle mit Indexnamen, Tabelle und Spalte wie in Anforderung

```
>>> #indices;
```

#indices.name	#indices.table	#indices.column
professoren_PersNr	professoren	PersNr
studenten_MatrNr	studenten	MatrNr
studenten_Name	studenten	Name
vorlesungen_VorlNr	vorlesungen	VorlNr
vorlesungen_gelesenVon	vorlesungen	gelesenVon

Index Seek

```
1 def __init__(..):
2     ....
3     self.index_column = self.schema.get_simple_column_name(index_column)
4     self.index = table_service.retrieve_index(self.table_name, self.index_column)
5     self.comparison_value = self._consume_condition()
6
7 def _get_index_records(self):
8     key = self.comparison_value
9     if key in self.index:
10         result = self.index[key]
11     else:
12         result = []
13     return result
```

Anwendung Index Seek

- Wie bei Selection Pushdown rekursives durchgehen aller Selections
- Prüft alle aufeinanderfolgenden Selections vor Table Scan, ob IndexSeek geeignet ist
- Wählt am besten geeignete Selection, ersetzt diese mit Kind und ersetzt Table Scan mit Index Seek

```
1 execution_plan = _node_access_helper(  
2     execution_plan, _split_selections, Selection)  
3 execution_plan = _node_access_helper(  
4     execution_plan, _selection_push_down(), Selection)  
5 execution_plan = _node_access_helper(  
6     execution_plan, _apply_index_seek, Selection)  
7 execution_plan = _node_access_helper(  
8     execution_plan, _join_selections, Selection)
```

Optimizer – Selection Pushdown – Projection Operator

- Referenzierte Spalten müssen umbenannt werden
→ Aliase und Fully-Qualified-Namen
- Falls berechnete Spalten referenziert werden, kann nicht gepusht werden
→ z.B. bei `fullName = firstName + " " + lastName`
- Falls alle Bedingungen erfüllt werden, wird auf „Projection“ gepusht



