

# Questions on Indexes and Table Storage

## 1 True/False Questions

For each question below, circle either True or False. On your final exam, each correct answer will result in +1 point, each incorrect answer will result in -1 point, and each blank answer in 0 points. For this homework assignment, you can uncomment the following line in the tex file to view the answers:

`\printanswers`

and so these questions do not need to be submitted. You should still try to complete them, however, to check your understanding. Approximately 4/5 of these questions are answered in class, and the remaining 1/5 you'll have to refer to the postgres documentation / supplementary material for the answers.

### Hash Index

1. True    **FALSE**    A hash index can return columns in sorted order.
2. **TRUE**    False    A has index on a text column whose average length is 100s of characters will use less disk space than an equivalent btree index.
3. True    **FALSE**    A table can be CLUSTERed on a hash index.
4. **TRUE**    False    A hash index can be used for index scans.
5. **TRUE**    False    A hash index can be used to speed up a nested loop join.
6. True    **FALSE**    A hash index can be used to speed up a merge join.
7. True    **FALSE**    A hash index can be used to speed up a hash join.
8. True    **FALSE**    The main advantage of a hash index over a btree index is that a hash index can result in fewer TABLE page accesses.
9. **TRUE**    False    The main advantage of a hash index over a btree index is that a hash index can result in fewer INDEX page accesses.

### GIN Index

10. True    **FALSE**    The GIN index supports index only scans.
11. True    **FALSE**    The GIN index supports index scans.
12. **TRUE**    False    The GIN index supports bitmap scans.
13. True    **FALSE**    A table can be CLUSTERed on a GIN index.
14. True    **FALSE**    You have a SELECT query that returns hundreds of thousands of rows. Postgres is using a GIN index to speed up the query, but it is still taking a long time. The query could be sped up dramatically by adding a LIMIT clause to reduce the number of rows returned.
15. **TRUE**    False    A GIN index can created on multiple columns.
16. True    **FALSE**    A GIN index can be used to speed up merge joins if the join condition is constructed appropriately.
17. True    **FALSE**    A GIN index can be used to speed up hash joins if the join condition is constructed appropriately.
18. **TRUE**    False    A GIN index can be used to speed up nested loop joins if the join condition is constructed appropriately.

19. True    **FALSE**    A GIN index created on a **tsvector** stores information about the position of lexemes within the document.
20. True    **FALSE**    If Postgres crashes while a DELETE/INSERT/UPDATE statement is modifying a GIN index, the index becomes corrupted and must be regenerated from scratch.

## RUM Index

21. True    **FALSE**    The RUM index supports index only scans.
22. **TRUE**    False    The RUM index supports index scans.
23. **TRUE**    False    The RUM index supports bitmap scans.
24. **TRUE**    False    You have a SELECT query that returns hundreds of thousands of rows. Postgres is using a RUM index to speed up the query, but it is still taking a long time. The query could be sped up dramatically by adding a LIMIT clause to reduce the number of rows returned.
25. True    **FALSE**    A table can be CLUSTERed on a RUM index.
26. True    **FALSE**    A RUM index can be used to speed up merge joins if the join condition is constructed appropriately.
27. True    **FALSE**    A RUM index can be used to speed up hash joins if the join condition is constructed appropriately.
28. **TRUE**    False    A RUM index can be used to speed up nested loop joins if the join condition is constructed appropriately.
29. **TRUE**    False    The RUM index uses more disk space than the GIN index.
30. **TRUE**    False    A RUM index created on a **tsvector** stores information about the position of lexemes within the document.
31. **TRUE**    False    RUM indexes do not support the **fastupdate** index creation parameter, and therefore inserting on a RUM index is slower than on a GIN index.
32. True    **FALSE**    If Postgres crashes while a DELETE/INSERT/UPDATE statement is modifying a RUM index, the index becomes corrupted and must be regenerated from scratch.

## Unicode

33. **TRUE**    False    PostgreSQL's implementation of UTF-8 is compliant with the Unicode standard.
34. **TRUE**    False    Emojis can be stored in a TEXT column if the database is using UTF-8 encodings.
35. **TRUE**    False    Given any string in NFC form, normalizing to NFD and back to NFC is guaranteed to be an idempotent operation (i.e. you will get the same string back.)
36. True    **FALSE**    Given any string in NFKD form, normalizing to NFD and back to NFKD is guaranteed to be an idempotent operation (i.e. you will get the same string back.)
37. **TRUE**    False    Given the string "César Chávez", an NFC-normalized UTF-8 encoding will require fewer bytes than a NFD-normalized UTF-8 encoding.
38. **TRUE**    False    Postgres can compress TEXT columns no matter what language is contained.
39. True    **FALSE**    Postgres will automatically normalize all text into NFC form.
40. **TRUE**    False    All characters from the Klingon writing system can be represented in Unicode.
41. True    **FALSE**    NFD is a system for encoding Unicode code points as bytes.

- 42. True    **FALSE**    ANSI is a system for encoding Unicode code points as bytes.
- 43. **TRUE**    False    ASCII is a system for encoding Unicode code points as bytes.
- 44. **TRUE**    False    UTF-8 is a system for encoding Unicode code points as bytes.
- 45. **TRUE**    False    UTF-16 is a system for encoding Unicode code points as bytes.

#### Full Text Search

- 46. **TRUE**    False    Postgresql's built-in `to_tsvector` function has support for the Arabic language.
- 47. True    **FALSE**    Postgresql's built-in `to_tsvector` function has support for the Korean language.
- 48. **TRUE**    False    Postgresql's built-in `to_tsvector` function has support for the Spanish language.
- 49. True    **FALSE**    Postgresql's built-in `pg_trgm` allows searching Chinese text.
- 50. **TRUE**    False    A GIN index built using bigrams generated from the `pg_bigm` extension will be crash-safe (i.e. the index will be valid if postgres crashes during an INSERT/UPDATE/DELETE operation).
- 51. True    **FALSE**    A pgroonga index will be crash safe (i.e. the index will be valid if postgres crashes during an INSERT/UPDATE/DELETE operation).
- 52. **TRUE**    False    A GIN index created with the `gin_trgm_ops` operator class supports faster searches using the LIKE operator.
- 53. True    **FALSE**    A GIN index created with the `gin_trgm_ops` operator class supports faster searches using the @@ operator.
- 54. True    **FALSE**    A GIN index created with the `tsvector_ops` operator class supports faster searches using the LIKE operator.
- 55. **TRUE**    False    A GIN index created with the `tsvector_ops` operator class supports faster searches using the @@ operator.