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 complies 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).