

Einführung in Datenbanken

— Übungsblatt 5 (Information Schema) —

Ihre Lösungen laden Sie bitte in die Übungsplattform in StudIP hoch ([StudIP-Eintrag der Vorlesung], Reiter „Übungsplattform“, dann auf „Weiter zu ... Übungsplattform“).

Einsendeschluss ist Montag, der 22.11.2021, 18⁰⁰.

Hausaufgabe 5 (16 Punkte)

Mit den einfachen SQL-Anfragen auf diesem Übungsblatt soll das „Information Schema“ vorgestellt werden, das Teil des SQL-Standards ist. Das „Information Schema“ ist eine Art standardisierter Systemkatalog. Es ist nicht in allen DBMS implementiert, aber in PostgreSQL und z.B. auch in Microsoft SQL Server und MySQL/MariaDB.

Die Anfragen sollten relativ einfach sein, aber es ist vielleicht gewöhnungsbedürftig, dass hier z.B. Tabellennamen als Daten auftauchen („Meta-Daten“: Daten über Daten). Im Adminer können Sie eine Liste aller Tabellen sehen, und für jede Tabelle die Spalten und weitere Daten — insofern besteht nicht sofort das Bedürfnis, auch Anfragen an den Systemkatalog zu stellen. Man muss sich aber fragen, woher der Adminer die Daten bekommt. Außerdem gibt es viele Daten, die der Adminer nicht anzeigt, die aber im Systemkatalog vorhanden sind. Speziellere Suchen werden von solchen graphischen Schnittstellen auch nicht unterstützt: Es wäre z.B. möglich, dass Sie sich nicht mehr genau an den Namen einer Tabelle erinnern können (und es sehr viele Tabellen gibt). Mit Anfragen an den Systemkatalog könnten Sie nach Tabellen suchen, deren Name eine bestimmte Teilzeichenkette enthält.

Öffnen Sie den Adminer:

```
[https://dbs.informatik.uni-halle.de/edb?pgsql=db&
  username=student_gast&db=postgres&ns=]
```

Zwar gibt es das „Information Schema“ auch im Menu zur Schema-Auswahl, aber Suchen nach Tabellen-Namen führt man oft aus, während man eigentlich in einem Anwendungsschema arbeitet. Für Teil d) brauchen Sie z.B. das Schema „empdept_public“. Wählen Sie z.B. dieses Schema. Um dann auf Tabellen im „Information Schema“ zugreifen zu können, müssen Sie sich explizit mit dem Schema-Präfix „information_schema.“ vor den Tabellennamen auf dieses Schema beziehen.

Die PostgreSQL-Dokumentation zum Information Schema finden Sie hier:

[<https://www.postgresql.org/docs/9.2/information-schema.html>]

Information Schema in weiteren Systemen:

- MySQL:

[<https://dev.mysql.com/doc/refman/8.0/en/information-schema.html>]

- MariaDB:

[<https://mariadb.com/kb/en/information-schema-tables/>]

- Microsoft SQL Server:

[<https://docs.microsoft.com/en-us/sql/relational-databases/system-information-schema-views/system-information-schema-views-transact-sql>]

Einige der Tabellen (z.B. `information_schema.check_constraints`) sind leer, weil sie nur Angaben über eigene Tabellen enthalten. Dem Nutzer `student_gast` gehört aber keine einzige Tabelle der Datenbank.

Anfragen, die Syntaxfehler enthalten, werden höchstwahrscheinlich mit 0 Punkten bewertet. Testen Sie Ihre Anfragen und nutzen Sie notfalls rechtzeitig das Forum in StudIP, um Hilfe zu bekommen.

- a) (4 Punkte) Welche Tabellen gibt es in der Vitamin-Datenbank? Geben Sie nur die Tabellen-Namen aus. Nutzen Sie die Tabelle `information_schema.tables`. In der Spalte `table_schema` muss der Wert `vit_public` stehen. Der gesuchte Tabellen-Name steht in der Spalte `table_name`.

Das Ergebnis sollte so aussehen:

<code>table_name</code>
<code>zutat</code>
<code>praeparat</code>
<code>inhalt</code>
<code>stoff</code>

Die Reihenfolge der Zeilen ist nicht vorgeschrieben (Relationen sind Mengen bzw. in SQL Multimengen von Tupeln). Auch die Spalten-Überschriften können anders sein als in der Beispiel-Ausgabe gezeigt. Später werden möglicherweise eine bestimmte Sortierung oder vorgegebene Spalten-Überschriften verlangt, aber die notwendigen Konstrukte wurden in der Vorlesung noch nicht eingeführt.

- b) (4 Punkte) Finden Sie alle Tabellen im Information Schema, die die Teil-Zeichenkette `column` im Namen haben. Bedenken Sie, dass auch die Tabellen des „Information Schema“ selbst im „Information Schema“ Systemkatalog eingetragen sind. Sie benötigen wieder die Tabelle `information_schema.tables`.

Für diese Anfrage müssen Sie zwei Bedingungen mit `AND` verknüpfen:

- Die gesuchten Tabellen sollen aus dem `information_schema` sein.
- Der Tabellen-Name soll `column` als Teil-Zeichenkette enthalten.

Die `AND`-Verknüpfung stellt sicher, dass nur Zeilen betrachtet werden, die beide Bedingungen erfüllen (logisches „und“).

Das Ergebnis sollte so aussehen:

table_name
<code>column_column_usage</code>
<code>column_domain_usage</code>
<code>column_privileges</code>
<code>role_column_grants</code>
<code>column_udt_usage</code>
<code>columns</code>
<code>constraint_column_usage</code>
<code>routine_column_usage</code>
<code>key_column_usage</code>
<code>triggered_update_columns</code>
<code>view_column_usage</code>
<code>column_options</code>

- c) (4 Punkte) Geben Sie eine Liste aller Spalten der Tabelle „`praeparat`“ von einem `numeric`-Typ aus, zusammen mit der Position der Spalte in der Tabelle (das ist die `ordinal_position`) und dem genauen Typ, also inklusive den beiden Typ-Parametern `numeric_precision` und `numeric_scale`. Der Typ soll als Zeichenkette in der üblichen Form ausgegeben werden, also z.B. `NUMERIC(4,2)` für den Preis. Da in der Vorlesung noch keine Fallunterscheidungen besprochen wurden, ist in Ordnung, dass die Anzahl 0 von Nachkommastellen explizit angegeben wird, also z.B. `NUMERIC(8,0)` für die PZN. Sie finden die notwendigen Daten in `information_schema.columns`.

Sie müssen bei dieser Aufgabe mehrere Bedingungen und `WHERE` mit `AND` verknüpfen (d.h. Sie suchen nur Zeilen, die alle diese Bedingungen erfüllen):

- Das `table_schema` soll `vit_public` sein.
- Der `table_name` ist `praeparat`.
- Der Datentyp ist `numeric`.

Das Ergebnis sollte so aussehen:

ordinal_position	column_name	?column?
1	pid	NUMERIC(3,0)
4	pzn	NUMERIC(8,0)
5	anz	NUMERIC(3,0)
7	tagesdosis	NUMERIC(2,0)
8	gewicht	NUMERIC(4,1)
9	preis	NUMERIC(4,2)

Der etwas komische dritte Spaltenname wurde von PostgreSQL gewählt, weil es für den Wertausdruck dort nicht automatisch einen guten Spalten-Namen bestimmen konnte. Wenn Sie wollen, können Sie mit dem Zusatz `AS "Datentyp"` einen sinnvollen Spaltennamen vorgeben. Sie müssen das aber nicht.

- d) (4 Punkte) Für diese Aufgabe wählen Sie bitte das Schema „empdept_public“. In diesem Schema gibt es eine Tabelle „emp“ mit Informationen über Angestellte („employees“) einer Firma. Der aktuelle Chef, Mr. King, wurde am 17. November 1981 eingestellt. Gesucht sind jetzt alle Angestellten, die nach ihm eingestellt wurden (deren Einstellung also von Mr. King veranlasst oder genehmigt wurde). In der Spalte „hiredate“ (vom Typ DATE) muss dann also ein Wert nach dem 17.11.1981 stehen. Geben Sie den Angestellten-Namen (ename), das Einstellungsdatum (hiredate) und die Differenz in Tagen zum 17.11.1981 aus.

Das Ergebnis sollte so aussehen:

ename	hiredate	?column?
SCOTT	1982-12-09	387
ADAMS	1983-01-12	421
JAMES	1981-12-03	16
FORD	1981-12-03	16
MILLER	1982-01-23	67

Falls Sie sich noch ein paar interessante Tabellen des Information Schema anschauen wollen, probieren Sie z.B. diese:

- `information_schema.sql_implementation_info`
- `information_schema.sql_features`
- `information_schema.sql_sizing`
- `information_schema.applicable_roles`
- `information_schema.table_privileges`
- `information_schema.key_column_usage`
(Leider ist die zugehörige Tabelle `table_constraints` bei uns leer, weil dort nur Constraints auf eigenen Tabellen gelistet werden. Das passt nicht zusammen.)