Hinweise

Ergebnis (0: nicht äquivalent, 1: äquivalent, (pm=0..1): tw. Äquivalent, error: Fehler → System kann Querys nicht vergleichen)

pm = partial marking (z.B. zu 85% korrekt)

pm, sonst 0 → falls Verfahren partielle Bewertung unterstützt wird pm erwartet, ansonsten nicht äquivalent erwartet

Muster (Name = x, pm = true/false)

tats. = tatsächlich

				Ergebnis		
Mutations-Kategorie/ SQL-Feature	Nr.	Query (Musterlösung)	Test-Query (Kandidat)	erwartet	tats.	Anmerkung
columns select vs. Select *	1	SELECT * FROM Album;	SELECT AlbumId, Title, ArtistId FROM Album;	1		
	2	SELECT AlbumId, Title, ArtistId FROM Album;	SELECT * FROM Album;	1		
where: same semantic, minimal syntax change	3	SELECT TrackId FROM PlaylistTrack WHERE TrackId < 100;	SELECT TrackId FROM PlaylistTrack WHERE TrackId <= 99;	1		
Semicolon ;	4	SELECT Name FROM Genre;	SELECT Name FROM Genre	1		
syntax sql keywords (upper, lower case)	5	SELECT Name FROM MediaType;	select Name FrOm MediaType;	1		
column count differing	6	SELECT InvoiceDate, BillingAddress, BillingCity FROM Invoice;	SELECT BillingAddress, BillingCity FROM Invoice;	pm, sonst 0		
	7	SELECT BillingAddress, BillingCity FROM Invoice;	SELECT InvoiceDate, BillingAddress, BillingCity FROM Invoice;	pm, sonst 0		
column ordering	8		SELECT BillingCity, BillingAddress FROM Invoice;	pm, sonst 0		
order by	9	SELECT TrackId, Name, GenreId from Track ORDER BY GenreId ASC;	SELECT TrackId, Name, GenreId from Track ORDER BY GenreId;	1		
	10	SELECT TrackId, Name, GenreId from Track ORDER BY GenreId DESC;	SELECT TrackId, Name, GenreId from Track;	pm, sonst 0		
	11		SELECT TrackId, Name, GenreId from Track ORDER BY GenreId ASC;	pm, sonst 0		
	12	SELECT Name, MediaTypeId FROM Track	SELECT Name, MediaTypeId FROM Track WHERE MediaTypeId > 2 and MediaTypeId > 1;	1		

Redundanzen	13	WHERE MediaTypeId > 2;	SELECT Name, MediaTypeId FROM Track WHERE MediaTypeId > 2 or MediaTypeId > 1;	pm, sonst 0		
Sub-Query/Join	14	select Track.TrackId, Track.Name, Track.Composer from Track join (select MediaType.MediaTypeId, MediaType.Name from MediaType where MediaType.Name = 'AAC audio file') as t_MediaType ON Track.MediaTypeId = t_MediaType.MediaTypeId;	select Track.TrackId, Track.Name, Track.Composer from Track join (select MediaType.MediaTypeId, MediaType.Name from MediaType) as t_MediaType ON Track.MediaTypeId = t_MediaType.MediaTypeId;	pm, sonst 0		
	15		select Track.TrackId, Track.Name, Track.Composer from Track Join MediaType ON Track.MediaTypeId = MediaType.MediaTypeId;	pm, sonst 0		
	16		select Track.TrackId, Track.Name, Track.Composer from Track join MediaType ON Track.MediaTypeId = MediaType.MediaTypeId WHERE MediaType.Name = 'AAC audio file';	1		
	17		select Track.TrackId, Track.Name, Track.Composer from Track join MediaType ON Track.MediaTypeId = MediaType.MediaTypeId AND MediaType.Name = 'AAC audio file';	1		

	18		select Track.TrackId, Track.Name, Track.Composer from Track, MediaType where Track.MediaTypeId = MediaType.MediaTypeId and (MediaType.Name = 'AAC audio file')	1		
distinct	19	SELECT DISTINCT PlaylistId FROM PlaylistTrack;	SELECT PlaylistId FROM PlaylistTrack;	pm, sonst 0		

where, and	20	SELECT * FROM Employee
		WHERE Title = 'Sales Support Agent';
			SELECT * FROM Employee;	pm, sonst 0		keine Where-Klausel

	21		SELECT * FROM Employee
			WHERE ReportsTo = 2;
				pm, sonst 0		zufällig gleiches Ergebnis-Set aber andere Syntax & Semantik

	22	SELECT Email FROM Employee
		WHERE City = 'Calgary' AND ReportsTo = 1;
			SELECT Email FROM Employee
			WHERE City = 'Calgary';
				pm, sonst 0		

or	23	SELECT LastName, FirstName, City FROM Employee
		WHERE City = 'Edmonton' OR City = 'Lethbridge';
			SELECT LastName, FirstName, City FROM Employee
			WHERE City = 'Edmonton';
				pm, sonst 0		

	24		SELECT LastName, FirstName, City FROM Employee
			WHERE City = 'Edmonton' OR City = 'Calgary';
				pm, sonst 0		

not	25	SELECT LastName, FirstName, City FROM Employee
		WHERE NOT City = 'Calgary';
			SELECT LastName, FirstName, City FROM Employee
			WHERE City = 'Lethbridge' OR City = 'Edmonton';
				pm, sonst 0		andere Syntax & Semantik → Test für Verfahren mit pragmatischen Ansatz

	26		SELECT LastName, FirstName, City FROM Employee
			WHERE NOT City = 'Edmonton';
				pm, sonst 0		

limit	27	SELECT LastName, FirstName, City FROM Employee
		LIMIT 3;
			SELECT LastName, FirstName, City FROM Employee
			LIMIT 4;
				pm, sonst 0		

	28		SELECT LastName, FirstName, City FROM Employee;	0		

min, max	29	SELECT MIN(UnitPrice) FROM InvoiceLine;	SELECT UnitPrice FROM InvoiceLine LIMIT 1;	0		Verfahren mit pragmatischen Ansatz
	30		SELECT MAX(UnitPrice) FROM InvoiceLine;	0		
	31	SELECT MAX(UnitPrice) FROM InvoiceLine;	SELECT UnitPrice FROM InvoiceLine LIMIT 1;	0		
count	32	SELECT COUNT(UnitPrice) FROM InvoiceLine WHERE UnitPrice = 0.99;	SELECT UnitPrice FROM InvoiceLine WHERE UnitPrice = 0.99;	0		
avg	33	SELECT avg(Total) FROM Invoice;	SELECT Total FROM Invoice;	0		
sum	34	SELECT sum(Total) FROM Invoice;		0		
like	35	SELECT ArtistId, Name FROM Artist WHERE Name LIKE '%Metal%';	SELECT ArtistId, Name FROM Artist;	0		
	36	SELECT ArtistId, Name FROM Artist WHERE Name LIKE 'Sant%';	SELECT ArtistId, Name FROM Artist WHERE Name LIKE 'San%';	pm, sonst 0		
	37	SELECT ArtistId, Name FROM Artist WHERE Name LIKE 'M%o';	SELECT ArtistId, Name FROM Artist WHERE Name LIKE 'M%';	pm, sonst 0		
	38	SELECT ArtistId, Name FROM Artist WHERE Name LIKE '_ %';	SELECT ArtistId, Name FROM Artist WHERE Name LIKE '_%';	pm, sonst 0		
in	39	SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice WHERE BillingCountry IN ('USA', 'Canada', 'Brazil');	SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice;	0		
	40		SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice WHERE BillingCountry IN ('USA', 'Brazil');	pm, sonst 0		
between	41	SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice WHERE InvoiceDate between '2012-06-04 00:00:00' AND '2012-07-28 00:00:00';	SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice;	0		
	42		SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice WHERE InvoiceDate between '2012-05-04 00:00:00' AND '2012-07-28 00:00:00';	pm, sonst 0		
	43	SELECT InvoiceId, CustomerId FROM Invoice WHERE InvoiceId between 50 AND 100;	SELECT InvoiceId, CustomerId FROM Invoice WHERE InvoiceId between 0 AND 120;	pm, sonst 0		
column alias	44	SELECT Name AS Genres FROM Genre;	SELECT Name AS Genre FROM Genre;	pm, sonst 0		

union	45	SELECT City FROM Customer
		UNION
		SELECT City FROM Employee;
			SELECT City FROM Customer;	0		

	46		SELECT City FROM Customer
			UNION
			SELECT BillingCity FROM Invoice;
				pm, sonst 0		

	47	SELECT City FROM Customer
		UNION ALL
		SELECT City FROM Employee;
			SELECT City FROM Customer
			UNION
			SELECT City FROM Employee;
				pm, sonst 0		

group by, having	~~48~~	~~SELECT InvoiceId, CustomerId, count(CustomerId) FROM Invoice GROUP BY CustomerId;~~	~~SELECT InvoiceId, CustomerId, count(CustomerId) FROM Invoice;~~	~~0~~		aus Kompatibilitätsgründen entfernt

	49		SELECT InvoiceId, CustomerId,
			count(CustomerId) FROM Invoice
			GROUP BY InvoiceId;
				pm, sonst 0		

	50	SELECT CustomerId, count(CustomerId) FROM Invoice
		group by CustomerId
		having count(CustomerId) < 7;
			SELECT CustomerId, count(CustomerId) FROM Invoice
			group by CustomerId;
				0		

	51		SELECT CustomerId, count(CustomerId) FROM Invoice
			group by CustomerId
			having count(CustomerId) > 1;
				pm, sonst 0		

exists	52	SELECT TrackID, Name, GenreID
		FROM Track
		WHERE EXISTS
		(SELECT GenreID FROM Genre WHERE Track.GenreId = Genre.GenreId AND (Genre.Name = 'Reggae' OR Genre.Name = 'Jazz'));
			SELECT TrackID, Name, GenreID
			FROM Track
			WHERE EXISTS
			(SELECT GenreID FROM Genre WHERE Track.GenreId = Genre.GenreId AND Genre.Name = 'Reggae' OR Genre.Name = 'Jazz');
				pm, sonst 0		

	53		SELECT TrackID, Name, GenreID
			FROM Track;
				0		

	54		SELECT TrackID, Name, GenreID
			FROM Track
			WHERE GenreID IN
			(SELECT GenreID FROM Genre WHERE Track.GenreId = Genre.GenreId AND (Genre.Name = 'Reggae' OR Genre.Name = 'Jazz'));
				1		

Inner, left, self join	55	SELECT PlaylistTrack.PlaylistId, Track.TrackId, Track.Name, Genre.Name FROM ((PlaylistTrack INNER JOIN Track ON PlaylistTrack.TrackId = Track.TrackId) INNER join Genre ON Genre.GenreId = Track.GenreId);	SELECT PlaylistTrack.PlaylistId, Track.TrackId, Track.Name, Genre.Name FROM ((PlaylistTrack JOIN Track ON PlaylistTrack.TrackId = Track.TrackId) INNER join Genre ON Track.GenreId = Genre.GenreId);	1		

	56	SELECT Customer.CustomerId, Invoice.InvoiceId
		FROM Customer LEFT JOIN Invoice
		ON Customer.CustomerId = Invoice.InvoiceId
			SELECT Customer.CustomerId, Invoice.InvoiceId
			FROM Customer LEFT OUTER JOIN Invoice
			ON Customer.CustomerId = Invoice.InvoiceId
				1		

	57	SELECT A.FirstName AS CustomerName1, B.FirstName AS CustomerName2, A.City
		FROM Customer A, Customer B
		WHERE A.CustomerID != B.CustomerID
		AND A.City = B.City
		ORDER BY A.City;
			SELECT X.FirstName AS CustomerName1, Y.FirstName AS CustomerName2, Y.City
			FROM Customer X, Customer Y
			WHERE Y.CustomerID <> X.CustomerID
			AND X.City = Y.City
			ORDER BY Y.City;
				1		

to much joins	58	SELECT PlaylistTrack.PlaylistId, Track.TrackId, Track.Name FROM
		(PlaylistTrack INNER JOIN Track ON PlaylistTrack.TrackId = Track.TrackId
		);
			SELECT PlaylistTrack.PlaylistId, Track.TrackId, Track.Name, Genre.Name FROM
			((PlaylistTrack INNER JOIN Track ON PlaylistTrack.TrackId = Track.TrackId)
			INNER join
			Genre ON Genre.GenreId = Track.GenreId
			);
				0		

Anmerkung Query 48:

lösung: SELECT InvoiceId, CustomerId, count(CustomerId) FROM Invoice GROUP BY CustomerId;

Bsp. kandidat: SELECT InvoiceId, CustomerId, count(CustomerId) FROM Invoice;

Erwartetes Ergebnis: 0

Test wurde entfernt, da nicht alle Tools/DB-Systeme die Kandidat-Query unterstützt haben, nur SQLite.

Deshalb aus Gründen der Konsistenz/Kompatibilität Test 48 entfernt

Hinweise

Ergebnis (0: nicht äquivalent, 1: äquivalent, (pm=0..1): tw. Äquivalent, error: Fehler → System kann Querys nicht vergleichen)

pm = partial marking (z.B. zu 85% korrekt)

pm, sonst 0 → falls Verfahren partielle Bewertung unterstützt wird pm erwartet, ansonsten nicht äquivalent erwartet

Name = testSQL, pm = false

				Ergebnis	
Mutations-Kategorie/ SQL-Feature	Nr.	Query (Musterlösung)	Test-Query (Kandidat)	erwartet	tats.
columns select vs. Select *	1	SELECT * FROM Album;	SELECT AlbumId, Title, ArtistId FROM Album;	1	1
	2	SELECT AlbumId, Title, ArtistId FROM Album;	SELECT * FROM Album;	1	1
where: same semantic, minimal syntax change	3	SELECT TrackId FROM PlaylistTrack WHERE TrackId < 100;	SELECT TrackId FROM PlaylistTrack WHERE TrackId <= 99;	1	1
Semicolon ;	4	SELECT Name FROM Genre;	SELECT Name FROM Genre	1	1
syntax sql keywords (upper, lower case)	5	SELECT Name FROM MediaType;	select Name FrOm MediaType;	1	1
column count differing	6	SELECT InvoiceDate, BillingAddress, BillingCity FROM Invoice;	SELECT BillingAddress, BillingCity FROM Invoice;	0	0
	7	SELECT BillingAddress, BillingCity FROM Invoice;	SELECT InvoiceDate, BillingAddress, BillingCity FROM Invoice;	0	0
column ordering	8		SELECT BillingCity, BillingAddress FROM Invoice;	0	1
order by	9	SELECT TrackId, Name, GenreId from Track ORDER BY GenreId ASC;	SELECT TrackId, Name, GenreId from Track ORDER BY GenreId;	1	1
	10	SELECT TrackId, Name, GenreId from Track ORDER BY GenreId DESC;	SELECT TrackId, Name, GenreId from Track;	0	1
	11		SELECT TrackId, Name, GenreId from Track ORDER BY GenreId ASC;	0	1
	12	SELECT Name, MediaTypeId FROM Track	SELECT Name, MediaTypeId FROM Track WHERE MediaTypeId > 2 and MediaTypeId > 1;	1	1

Redundanzen	13	WHERE MediaTypeId > 2;	SELECT Name, MediaTypeId FROM Track WHERE MediaTypeId > 2 or MediaTypeId > 1;	0	0
Sub-Query/Join	14	select Track.TrackId, Track.Name, Track.Composer from Track join (select MediaType.MediaTypeId, MediaType.Name from MediaType where MediaType.Name = 'AAC audio file') as t_MediaType ON Track.MediaTypeId = t_MediaType.MediaTypeId;	select Track.TrackId, Track.Name, Track.Composer from Track join (select MediaType.MediaTypeId, MediaType.Name from MediaType) as t_MediaType ON Track.MediaTypeId = t_MediaType.MediaTypeId;	0	0
	15		select Track.TrackId, Track.Name, Track.Composer from Track Join MediaType ON Track.MediaTypeId = MediaType.MediaTypeId;	0	0
	16		select Track.TrackId, Track.Name, Track.Composer from Track join MediaType ON Track.MediaTypeId = MediaType.MediaTypeId WHERE MediaType.Name = 'AAC audio file';	1	1
	17		select Track.TrackId, Track.Name, Track.Composer from Track join MediaType ON Track.MediaTypeId = MediaType.MediaTypeId AND MediaType.Name = 'AAC audio file';	1	1

	18		select Track.TrackId, Track.Name, Track.Composer from Track, MediaType where Track.MediaTypeId = MediaType.MediaTypeId and (MediaType.Name = 'AAC audio file')	1	1
distinct	19	SELECT DISTINCT PlaylistId FROM PlaylistTrack;	SELECT PlaylistId FROM PlaylistTrack;	0	0

where, and	20	SELECT * FROM Employee
		WHERE Title = 'Sales Support Agent';
			SELECT * FROM Employee;	0	0

	21		SELECT * FROM Employee
			WHERE ReportsTo = 2;
				0	1

	22	SELECT Email FROM Employee
		WHERE City = 'Calgary' AND ReportsTo = 1;
			SELECT Email FROM Employee
			WHERE City = 'Calgary';
				0	0

or	23	SELECT LastName, FirstName, City FROM Employee
		WHERE City = 'Edmonton' OR City = 'Lethbridge';
			SELECT LastName, FirstName, City FROM Employee
			WHERE City = 'Edmonton';
				0	0

	24		SELECT LastName, FirstName, City FROM Employee
			WHERE City = 'Edmonton' OR City = 'Calgary';
				0	0

not	25	SELECT LastName, FirstName, City FROM Employee
		WHERE NOT City = 'Calgary';
			SELECT LastName, FirstName, City FROM Employee
			WHERE City = 'Lethbridge' OR City = 'Edmonton';
				0	1

	26		SELECT LastName, FirstName, City FROM Employee
			WHERE NOT City = 'Edmonton';
				0	0

limit	27	SELECT LastName, FirstName, City FROM Employee
		LIMIT 3;
			SELECT LastName, FirstName, City FROM Employee
			LIMIT 4;
				0	0

	28		SELECT LastName, FirstName, City FROM Employee;	0	0

min, max	29	SELECT MIN(UnitPrice) FROM InvoiceLine;	SELECT UnitPrice FROM InvoiceLine LIMIT 1;	0	0
	30		SELECT MAX(UnitPrice) FROM InvoiceLine;	0	0
	31	SELECT MAX(UnitPrice) FROM InvoiceLine;	SELECT UnitPrice FROM InvoiceLine LIMIT 1;	0	0
count	32	SELECT COUNT(UnitPrice) FROM InvoiceLine WHERE UnitPrice = 0.99;	SELECT UnitPrice FROM InvoiceLine WHERE UnitPrice = 0.99;	0	0
avg	33	SELECT avg(Total) FROM Invoice;	SELECT Total FROM Invoice;	0	0
sum	34	SELECT sum(Total) FROM Invoice;		0	0
like	35	SELECT ArtistId, Name FROM Artist WHERE Name LIKE '%Metal%';	SELECT ArtistId, Name FROM Artist;	0	0
	36	SELECT ArtistId, Name FROM Artist WHERE Name LIKE 'Sant%';	SELECT ArtistId, Name FROM Artist WHERE Name LIKE 'San%';	0	0
	37	SELECT ArtistId, Name FROM Artist WHERE Name LIKE 'M%o';	SELECT ArtistId, Name FROM Artist WHERE Name LIKE 'M%';	0	0
	38	SELECT ArtistId, Name FROM Artist WHERE Name LIKE '_ %';	SELECT ArtistId, Name FROM Artist WHERE Name LIKE '_%';	0	0
in	39	SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice WHERE BillingCountry IN ('USA', 'Canada', 'Brazil');	SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice;	0	0
	40		SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice WHERE BillingCountry IN ('USA', 'Brazil');	0	0
between	41	SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice WHERE InvoiceDate between '2012-06-04 00:00:00' AND '2012-07-28 00:00:00';	SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice;	0	0
	42		SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice WHERE InvoiceDate between '2012-05-04 00:00:00' AND '2012-07-28 00:00:00';	0	0
	43	SELECT InvoiceId, CustomerId FROM Invoice WHERE InvoiceId between 50 AND 100;	SELECT InvoiceId, CustomerId FROM Invoice WHERE InvoiceId between 0 AND 120;	0	0

column alias	44	SELECT Name AS Genres FROM Genre;	SELECT Name AS Genre FROM Genre;	0	0

union	45	SELECT City FROM Customer
		UNION
		SELECT City FROM Employee;
			SELECT City FROM Customer;	0	0

	46		SELECT City FROM Customer
			UNION
			SELECT BillingCity FROM Invoice;
				0	0

	47	SELECT City FROM Customer
		UNION ALL
		SELECT City FROM Employee;
			SELECT City FROM Customer
			UNION
			SELECT City FROM Employee;
				0	0

group by, having	49	SELECT InvoiceId, CustomerId, count(CustomerId) FROM Invoice GROUP BY CustomerId;	SELECT InvoiceId, CustomerId, count(CustomerId) FROM Invoice
			GROUP BY InvoiceId;
				0	0

	50	SELECT CustomerId, count(CustomerId) FROM Invoice
		group by CustomerId
		having count(CustomerId) < 7;
			SELECT CustomerId, count(CustomerId)
			FROM Invoice
			group by CustomerId;
				0	0

	51		SELECT CustomerId, count(CustomerId)
			FROM Invoice
			group by CustomerId
			having count(CustomerId) > 1;
				0	0

exists	52	SELECT TrackID, Name, GenreID
		FROM Track
		WHERE EXISTS
		(SELECT GenreID FROM Genre WHERE Track.GenreId = Genre.GenreId AND (Genre.Name = 'Reggae' OR Genre.Name = 'Jazz'));
			SELECT TrackID, Name, GenreID
			FROM Track
			WHERE EXISTS
			(SELECT GenreID FROM Genre WHERE Track.GenreId = Genre.GenreId AND Genre.Name = 'Reggae' OR Genre.Name = 'Jazz');
				0	0

	53		SELECT TrackID, Name, GenreID
			FROM Track;
				0	0

	54		SELECT TrackID, Name, GenreID
			FROM Track
			WHERE GenreID IN
			(SELECT GenreID FROM Genre WHERE Track.GenreId = Genre.GenreId AND (Genre.Name = 'Reggae' OR Genre.Name = 'Jazz'));
				1	1

Inner, left, self join	55	SELECT PlaylistTrack.PlaylistId, Track.TrackId, Track.Name, Genre.Name FROM ((PlaylistTrack INNER JOIN Track ON PlaylistTrack.TrackId = Track.TrackId) INNER join Genre ON Genre.GenreId = Track.GenreId);	SELECT PlaylistTrack.PlaylistId, Track.TrackId, Track.Name, Genre.Name FROM ((PlaylistTrack JOIN Track ON PlaylistTrack.TrackId = Track.TrackId) INNER join Genre ON Track.GenreId = Genre.GenreId);	1	error

	56	SELECT Customer.CustomerId, Invoice.InvoiceId
		FROM Customer LEFT JOIN Invoice
		ON Customer.CustomerId = Invoice.InvoiceId
			SELECT Customer.CustomerId, Invoice.InvoiceId
			FROM Customer LEFT OUTER JOIN Invoice
			ON Customer.CustomerId = Invoice.InvoiceId
				1	1

	57	SELECT A.FirstName AS CustomerName1, B.FirstName AS CustomerName2, A.City
		FROM Customer A, Customer B
		WHERE A.CustomerID != B.CustomerID
		AND A.City = B.City
		ORDER BY A.City;
			SELECT X.FirstName AS CustomerName1, Y.FirstName AS CustomerName2, Y.City
			FROM Customer X, Customer Y
			WHERE Y.CustomerID <> X.CustomerID
			AND X.City = Y.City
			ORDER BY Y.City;
				1	1

to much joins	58	SELECT PlaylistTrack.PlaylistId, Track.TrackId, Track.Name FROM
		(PlaylistTrack INNER JOIN Track ON PlaylistTrack.TrackId = Track.TrackId
		);
			SELECT PlaylistTrack.PlaylistId, Track.TrackId, Track.Name, Genre.Name FROM
			((PlaylistTrack INNER JOIN Track ON PlaylistTrack.TrackId = Track.TrackId)
			INNER join
			Genre ON Genre.GenreId = Track.GenreId
			);
				0	0

tats. = tatsächlich

Anmerkung	Compare
	0
	0
	0
	0
	0
Expected only the following column(s) to be selected: InvoiceDate, BillingAddress, BillingCity!	0
Expected only the following column(s) to be selected: BillingAddress, BillingCity!	0
tatsächlich inkorrekt? Reihenfolge Columns. Mail nachfragen?	1
	0
order by wird nicht ausreichend behandelt!	1
order by wird nicht ausreichend behandelt!	1
	0

Expected a total of 232 row(s) to be returned, instead got 469!	0
Expected a total of 11 row(s) to be returned, instead got 3503!	0
Expected a total of 11 row(s) to be returned, instead got 3503!	0
	0
	0

	0
Expected a total of 14 row(s) to be returned, instead got 8715!	0
keine Where-Klausel	0
zufällig gleiches Ergebnis-Set aber andere Syntax & Semantik	1
Expected a total of 2 row(s) to be returned, instead got 5!	0
Expected a total of 3 row(s) to be returned, instead got 1!	0
Expected a total of 3 row(s) to be returned, instead got 6!	0
zufällig gleiches Ergebnis-Set aber andere Syntax & Semantik → Test für Verfahren mit pragmatischen Ansatz	1
Expected a total of 3 row(s) to be returned, instead got 7!	0
Expected a total of 3 row(s) to be returned, instead got 4!	0
Expected a total of 3 row(s) to be returned, instead got 8!	0

zufällig gleiches Ergebnis-Set aber andere Syntax & Semantik → Test für Verfahren mit pragmatischen Ansatz (result: Expected only the following column(s) to be selected: MIN(UnitPrice)!)	0
Expected only the following column(s) to be selected: MIN(UnitPrice)!	0
Expected only the following column(s) to be selected: MAX(UnitPrice)!	0
SELECT UnitPrice FROM InvoiceLine WHERE UnitPrice = 0.99;	0
Expected a total of 1 row(s) to be returned, instead got 412!	0
Expected a total of 1 row(s) to be returned, instead got 412!	0
Expected a total of 1 row(s) to be returned, instead got 275!	0
Expected a total of 9 row(s) to be returned, instead got 10!	0
Expected a total of 3 row(s) to be returned, instead got 20!	0
Expected a total of 3 row(s) to be returned, instead got 275!	0
Expected a total of 182 row(s) to be returned, instead got 412!	0
Expected a total of 182 row(s) to be returned, instead got 126!	0
Expected a total of 13 row(s) to be returned, instead got 412!	0
Expected a total of 13 row(s) to be returned, instead got 20!	0
Expected a total of 51 row(s) to be returned, instead got 120!	0

Expected only the following column(s) to be selected: Genres!	0
Expected a total of 55 row(s) to be returned, instead got 59!	0
Expected a total of 55 row(s) to be returned, instead got 53! (gar kein test auf WELCHE relation???)	0
Expected a total of 67 row(s) to be returned, instead got 55!	0
Expected a total of 59 row(s) to be returned, instead got 412!	0
Expected a total of 1 row(s) to be returned, instead got 59!	0
Expected a total of 1 row(s) to be returned, instead got 59!	0
Expected a total of 188 row(s) to be returned, instead got 3503!	0
Expected a total of 188 row(s) to be returned, instead got 3503!	0
	0

Tool gibt zwar Fehlermeldung aus, Ergebnis-Tabelle scheint aber korrekt zu sein!!! → Tool evtl. anfällig für große Joins	1	
	0	
	0	
Expected only the following column(s) to be selected: PlaylistId, TrackId, Name!	0	
		%
Expected != actual:	6	10,5
error	1	1,8

Hinweise

Ergebnis (0: nicht äquivalent, 1: äquivalent, (pm=0..1): tw. Äquivalent, error: Fehler → System kann Querys nicht vergleichen)

pm = partial marking (z.B. zu 85% korrekt)

pm, sonst 0 → falls Verfahren partielle Bewertung unterstützt wird pm erwartet, ansonsten nicht äquivalent erwartet

Name = Cosette, pm = false

				Ergebnis	
Mutations-Kategorie/ SQL-Feature	Nr.	Query (Musterlösung)	Test-Query (Kandidat)	erwartet	tats.
columns select vs. Select *	1	SELECT * FROM Album;	SELECT AlbumId, Title, ArtistId FROM Album;	1	unknown
	2	SELECT AlbumId, Title, ArtistId FROM Album;	SELECT * FROM Album;	1	unknown
where: same semantic, minimal syntax change	3	SELECT TrackId FROM PlaylistTrack WHERE TrackId < 100;	SELECT TrackId FROM PlaylistTrack WHERE TrackId <= 99;	1	error

Semicolon ;	4	SELECT Name FROM Genre;	SELECT Name FROM Genre	1	error
syntax sql keywords (upper, lower case)	5	SELECT Name FROM MediaType;	select Name FrOm MediaType;	1	1
column count differing	6	SELECT InvoiceDate, BillingAddress, BillingCity FROM Invoice;	SELECT BillingAddress, BillingCity FROM Invoice;	0	0
	7	SELECT BillingAddress, BillingCity FROM Invoice;	SELECT InvoiceDate, BillingAddress, BillingCity FROM Invoice;	0	0
column ordering	8		SELECT BillingCity, BillingAddress FROM Invoice;	0	0
	9	SELECT TrackId, Name, GenreId from Track ORDER BY GenreId ASC;	SELECT TrackId, Name, GenreId from Track ORDER BY GenreId;	1	error

order by	10	SELECT TrackId, Name, GenreId from Track ORDER BY GenreId DESC;	SELECT TrackId, Name, GenreId from Track;	0	error
	11		SELECT TrackId, Name, GenreId from Track ORDER BY GenreId ASC;	0	error
Redundanzen	12	SELECT Name, MediaTypeId FROM Track WHERE MediaTypeId > 2;	SELECT Name, MediaTypeId FROM Track WHERE MediaTypeId > 2 and MediaTypeId > 1;	1	unknown
	13		SELECT Name, MediaTypeId FROM Track WHERE MediaTypeId > 2 or MediaTypeId > 1;	0	0
	14		select Track.TrackId, Track.Name, Track.Composer from Track join (select MediaType.MediaTypeId, MediaType.Name from MediaType) as t_MediaType ON Track.MediaTypeId = t_MediaType.MediaTypeId;	0	0
	15		select Track.TrackId, Track.Name, Track.Composer from Track Join MediaType ON Track.MediaTypeId = MediaType.MediaTypeId;	0	0

Sub-Query/Join	16	select Track.TrackId, Track.Name, Track.Composer from Track join (select MediaType.MediaTypeId, MediaType.Name from MediaType where MediaType.Name = 'AAC audio file') as t_MediaType ON Track.MediaTypeId = t_MediaType.MediaTypeId;	select Track.TrackId, Track.Name, Track.Composer from Track join MediaType ON Track.MediaTypeId = MediaType.MediaTypeId WHERE MediaType.Name = 'AAC audio file';	1	unknown
	17		select Track.TrackId, Track.Name, Track.Composer from Track join MediaType ON Track.MediaTypeId = MediaType.MediaTypeId AND MediaType.Name = 'AAC audio file';	1	unknown
	18		select Track.TrackId, Track.Name, Track.Composer from Track, MediaType where Track.MediaTypeId = MediaType.MediaTypeId and (MediaType.Name = 'AAC audio file')	1	unknown
distinct	19	SELECT DISTINCT PlaylistId FROM PlaylistTrack;	SELECT PlaylistId FROM PlaylistTrack;	0	0

where, and	20	SELECT * FROM Employee
		WHERE Title = 'Sales Support Agent';
			SELECT * FROM Employee;	0	0

	21		SELECT * FROM Employee
			WHERE ReportsTo = 2;
				0	0

	22	SELECT Email FROM Employee
		WHERE City = 'Calgary' AND ReportsTo = 1;
			SELECT Email FROM Employee
			WHERE City = 'Calgary';
				0	0

or	23	SELECT LastName, FirstName, City FROM Employee
		WHERE City = 'Edmonton' OR City = 'Lethbridge';
			SELECT LastName, FirstName, City FROM Employee
			WHERE City = 'Edmonton';
				0	0

	24		SELECT LastName, FirstName, City FROM Employee
			WHERE City = 'Edmonton' OR City = 'Calgary';
				0	0

not	25	SELECT LastName, FirstName, City FROM Employee
		WHERE NOT City = 'Calgary';
			SELECT LastName, FirstName, City FROM Employee
			WHERE City = 'Lethbridge' OR City = 'Edmonton';
				0	0

	26		SELECT LastName, FirstName, City FROM Employee
			WHERE NOT City = 'Edmonton';
				0	0

limit	27	SELECT LastName, FirstName, City FROM Employee
		LIMIT 3;
			SELECT LastName, FirstName, City FROM Employee
			LIMIT 4;
				0	error

	28		SELECT LastName, FirstName, City FROM Employee;	0	error
min, max	29	SELECT MIN(UnitPrice) FROM InvoiceLine;	SELECT UnitPrice FROM InvoiceLine LIMIT 1;	0	error
	30		SELECT MAX(UnitPrice) FROM InvoiceLine;	0	0
	31	SELECT MAX(UnitPrice) FROM InvoiceLine;	SELECT UnitPrice FROM InvoiceLine LIMIT 1;	0	error
count	32	SELECT COUNT(UnitPrice) FROM InvoiceLine WHERE UnitPrice = 0.99;	SELECT UnitPrice FROM InvoiceLine WHERE UnitPrice = 0.99;	0	error
avg	33	SELECT avg(Total) FROM Invoice;	SELECT Total FROM Invoice;	0	error
sum	34	SELECT sum(Total) FROM Invoice;		0	0
like	35	SELECT ArtistId, Name FROM Artist WHERE Name LIKE '%Metal%';	SELECT ArtistId, Name FROM Artist;	0	error
	36	SELECT ArtistId, Name FROM Artist WHERE Name LIKE 'Sant%';	SELECT ArtistId, Name FROM Artist WHERE Name LIKE 'San%';	0	error
	37	SELECT ArtistId, Name FROM Artist WHERE Name LIKE 'M%o';	SELECT ArtistId, Name FROM Artist WHERE Name LIKE 'M%';	0	error
	38	SELECT ArtistId, Name FROM Artist WHERE Name LIKE '_ %';	SELECT ArtistId, Name FROM Artist WHERE Name LIKE '_%';	0	error
	39	SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice	SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice;	0	error

in	40	FROM Invoice
		WHERE BillingCountry IN ('USA', 'Canada', 'Brazil');
			SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice
			WHERE BillingCountry IN ('USA', 'Brazil');
				0	error

between	41	SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice
		WHERE InvoiceDate between '2012-06-04 00:00:00' AND '2012-07-28 00:00:00';
			SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice;	0	error

	42		SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice
			WHERE InvoiceDate between '2012-05-04 00:00:00' AND '2012-07-28 00:00:00';
				0	error

	43	SELECT InvoiceId, CustomerId FROM Invoice
		WHERE InvoiceId between 50 AND 100;
			SELECT InvoiceId, CustomerId FROM Invoice
			WHERE InvoiceId between 0 AND 120;
				0	error

column alias	44	SELECT Name AS Genres FROM Genre;	SELECT Name AS Genre FROM Genre;	0	1

union	45	SELECT City FROM Customer
		UNION
		SELECT City FROM Employee;
			SELECT City FROM Customer;	0	error

	46		SELECT City FROM Customer
			UNION
			SELECT BillingCity FROM Invoice;
				0	error

	47	SELECT City FROM Customer
		UNION ALL
		SELECT City FROM Employee;
			SELECT City FROM Customer
			UNION
			SELECT City FROM Employee;
				0	error

	49	SELECT InvoiceId, CustomerId, count(CustomerId) FROM Invoice GROUP BY CustomerId;	SELECT InvoiceId, CustomerId, count(CustomerId) FROM Invoice
			GROUP BY InvoiceId;
				0	0

group by, having	50	SELECT CustomerId, count(CustomerId) FROM Invoice group by CustomerId having count(CustomerId) < 7;	SELECT CustomerId, count(CustomerId) FROM Invoice group by CustomerId;	0	0
	51		SELECT CustomerId, count(CustomerId) FROM Invoice group by CustomerId having count(CustomerId) > 1;	0	0
exists	52	SELECT TrackID, Name, GenreID FROM Track WHERE EXISTS (SELECT GenreID FROM Genre WHERE Track.GenreId = Genre.GenreId AND (Genre.Name = 'Reggae' OR Genre.Name = 'Jazz'));	SELECT TrackID, Name, GenreID FROM Track WHERE EXISTS (SELECT GenreID FROM Genre WHERE Track.GenreId = Genre.GenreId AND Genre.Name = 'Reggae' OR Genre.Name = 'Jazz');	0	unknown
	53		SELECT TrackID, Name, GenreID FROM Track;	0	0
	54		SELECT TrackID, Name, GenreID FROM Track WHERE GenreID IN (SELECT GenreID FROM Genre WHERE Track.GenreId = Genre.GenreId AND (Genre.Name = 'Reggae' OR Genre.Name = 'Jazz'));	1	error

Inner, left, self join	55	SELECT PlaylistTrack.PlaylistId, Track.TrackId, Track.Name, Genre.Name FROM ((PlaylistTrack INNER JOIN Track ON PlaylistTrack.TrackId = Track.TrackId) INNER join Genre ON Genre.GenreId = Track.GenreId);	SELECT PlaylistTrack.PlaylistId, Track.TrackId, Track.Name, Genre.Name FROM ((PlaylistTrack JOIN Track ON PlaylistTrack.TrackId = Track.TrackId) INNER join Genre ON Track.GenreId = Genre.GenreId);	1	unknown

	56	SELECT Customer.CustomerId, Invoice.InvoiceId
		FROM Customer LEFT JOIN Invoice
		ON Customer.CustomerId = Invoice.InvoiceId
			SELECT Customer.CustomerId, Invoice.InvoiceId
			FROM Customer LEFT OUTER JOIN Invoice
			ON Customer.CustomerId = Invoice.InvoiceId
				1	error

	57	SELECT A.FirstName AS CustomerName1, B.FirstName AS CustomerName2, A.City
		FROM Customer A, Customer B
		WHERE A.CustomerID != B.CustomerID
		AND A.City = B.City
		ORDER BY A.City;
			SELECT X.FirstName AS CustomerName1, Y.FirstName AS CustomerName2, Y.City
			FROM Customer X, Customer Y
			WHERE Y.CustomerID <> X.CustomerID
			AND X.City = Y.City
			ORDER BY Y.City;
				1	error

to much joins	58	SELECT PlaylistTrack.PlaylistId, Track.TrackId, Track.Name FROM
		(PlaylistTrack INNER JOIN Track ON PlaylistTrack.TrackId = Track.TrackId
		);
			SELECT PlaylistTrack.PlaylistId, Track.TrackId, Track.Name, Genre.Name FROM
			((PlaylistTrack INNER JOIN Track ON PlaylistTrack.TrackId = Track.TrackId)
			  INNER join
			  Genre ON Genre.GenreId = Track.GenreId
			);
				0	0

tats. = tatsächlich

Anmerkung	Compare
{u'rosette_result': u'UNSAT', u'rosette_log': u", u'coq_result': u'UNKNOWN', u'result': u'UNKNOWN', u'coq_log': u"}. out of time: https://github.com/uwdb/Cosette/issues/80	1
{u'rosette_result': u'UNSAT', u'rosette_log': u", u'coq_result': u'UNKNOWN', u'result': u'UNKNOWN', u'coq_log': u"}	1
{u'rosette_log': u'{"size":[1],"status":"UNSAT"}\ngenerated/VxNmtKgrTvHqx.rkt:17:15: trackid: unbound identifier in module\n in: trackid\n context...:\n /root/.racket/6.8/pkgs/rosette/rosette/base/form/module.rkt:16:0\n standard-module-name-resolver\n /Cosette/rosette/server.rkt:38:10\n', u'coq_result': u'ERROR', u'coq_log': u'Invalid generated Coq code. Please file an issue.', u'result': u'ERROR', u'rosette_result': u'ERROR', u'error_msg': u'Invalid generated Coq code. Please file an issue. \n \n {"size":[1],"status":"UNSAT"}\ngenerated/VxNmtKgrTvHqx.rkt:17:15: trackid: unbound identifier in module\n in: trackid\n context...:\n /root/.racket/6.8/pkgs/rosette/rosette/base/form/module.rkt:16:0\n standard-module-name-resolver\n /Cosette/rosette/server.rkt:38:10\n'}	1

{u'rosette_log': u'Syntax Error. \n ERROR: "(line 10, column 1):\\nunexpected \'v\'\\nexpecting \\"--\\", \\"/*\\", \\";\\" or end of input"', u'coq_result': u'ERROR', u'coq_log': u'Syntax Error. \n ERROR: "(line 10, column 1):\\nunexpected \'v\'\\nexpecting \\"--\\", \\"/*\\", \\";\\" or end of input"', u'result': u'ERROR', u'counterexamples': [], u'rosette_result': u'ERROR', u'error_msg': u'Syntax Error. \n ERROR: "(line 10, column 1):\\nunexpected \'v\'\\nexpecting \\"--\\", \\"/*\\", \\";\\" or end of input"'}	1
	0
{u'rosette_log': u'Rosette find an counterexample.', u'coq_result': u'UNKNOWN', u'coq_log': u'', u'result': u'NEQ', u'counterexamples': [{u'table-content': [[u'invoiceid', u'customerid', u'invoicedate', u'billingaddress', u'billingcity', u'billingstate', u'billingcountry', u'billingpostalcode', u'total'], [[[0, 0, 0, 0, 0, 0, 0, 0, 0], 1]]], u'table-name': u'invoice'}], u'rosette_result': u'NEQ'}	0
	0
u'invoicedate', u'billingaddress', u'billingcity', u'billingstate', u'billingcountry', u'billingpostalcode', u'total'], [[[0, 0, 0, 1, 0, 0, 0, 0, 0], 1]]], u'table-name': u'invoice'}], u'rosette_result': u'NEQ'}	0
{u'rosette_log': u'Syntax Error. \n ERROR: "(line 7, column 61):\\nunexpected \\"b\\"\\nexpecting \\"--\\" or \\"/*\\""', u'coq_result': u'ERROR', u'coq_log': u'Syntax Error. \n ERROR: "(line 7, column 61):\\nunexpected \\"b\\"\\nexpecting \\"--\\" or \\"/*\\""', u'result': u'ERROR', u'counterexamples': [], u'rosette_result': u'ERROR', u'error_msg': u'Syntax Error. \n ERROR: "(line 7, column 61):\\nunexpected \\"b\\"\\nexpecting \\"--\\" or \\"/*\\""'}	1

{u'rosette_log': u'Syntax Error. \n ERROR: "(line 7, column 61):\\nunexpected \\"b\\"\\nexpecting \\"--\\" or \\"/*\\""', u'coq_result': u'ERROR', u'coq_log': u'Syntax Error. \n ERROR: "(line 7, column 61):\\nunexpected \\"b\\"\\nexpecting \\"--\\" or \\"/*\\""', u'result': u'ERROR', u'counterexamples': [], u'rosette_result': u'ERROR', u'error_msg': u'Syntax Error. \n ERROR: "(line 7, column 61):\\nunexpected \\"b\\"\\nexpecting \\"--\\" or \\"/*\\""'}	1	
s.o.	1	
Two queries' equivalence is unknown. Solver runs out of time.	1	
{u'rosette_log': u'Rosette find an counterexample.', u'coq_result': u'UNKNOWN', u'coq_log': u'', u'result': u'NEQ', u'counterexamples': [{u'table-content': [[u'trackid', u'name', u'albumid', u'mediatypeid', u'genreid', u'composer', u'milliseconds', u'bytes', u'unitprice'], [[[0, 0, 0, 2, 0, 0, 0, 0, 0], 1]]], u'table-name': u'track'}], u'rosette_result': u'NEQ'}	0	
{u'rosette_log': u'Rosette find an counterexample.', u'coq_result': u'UNKNOWN', u'coq_log': u'', u'result': u'NEQ', u'counterexamples': [{u'table-content': [[u'mediatypeid', u'name'], [[[0, 0], 13]]], u'table-name': u'mediatype'}, {u'table-content': [[u'trackid', u'name', u'albumid', u'mediatypeid', u'genreid', u'composer', u'milliseconds', u'bytes', u'unitprice'], [[[0, 0, 0, 0, 0, 0, 0, 0, 0], 10]]], u'table-name': u'track'}], u'rosette_result': u'NEQ'}	0	Hinweis counter example korrekt/sinnvoll?
{u'rosette_log': u'Rosette find an counterexample.', u'coq_result': u'UNKNOWN', u'coq_log': u'', u'result': u'NEQ', u'counterexamples': [{u'table-content': [[u'mediatypeid', u'name'], [[[0, 0], 13]]], u'table-name': u'mediatype'}, {u'table-content': [[u'trackid', u'name', u'albumid', u'mediatypeid', u'genreid', u'composer', u'milliseconds', u'bytes', u'unitprice'], [[[0, 0, 0, 0, 0, 0, 0, 0, 0], 10]]], u'table-name': u'track'}], u'rosette_result': u'NEQ'}	0	

{u'rosette_result': u'UNSAT', u'rosette_log': u'', u'coq_result': u'UNKNOWN', u'result': u'UNKNOWN', u'coq_log': u''} (solver runs out of time)	1
{u'rosette_result': u'UNSAT', u'rosette_log': u'', u'coq_result': u'UNKNOWN', u'result': u'UNKNOWN', u'coq_log': u''} Two queries' equivalence is unknown. Solver runs out of time.	1
{u'rosette_result': u'UNSAT', u'rosette_log': u'', u'coq_result': u'UNKNOWN', u'result': u'UNKNOWN', u'coq_log': u''} Two queries' equivalence is unknown. Solver runs out of time.	1
Two queries are not equivalent. Counter Examples: (i.e., input tables that, when fed into your input queries, will return different results) Table playlisttrack playlistidtrackid 00 00	0

keine Where-Klausel. Ergebnis: Two queries are not equivalent. Counter Examples: (i.e., input tables that, when fed into your input queries, will return different results) Table employee employeeidlastnamefirstnametitlereportstobirthdatehiredateaddrescitystatecountrypostalcodephonefaxemail 000000000000000 ??	0
zufällig gleiches Ergebnis-Set aber andere Syntax & Semantik Two queries are not equivalent. Counter Examples: (i.e., input tables that, when fed into your input queries, will return different results) Table employee employeeid lastname firstname title reportsto birthdate hiredate address city state country postalcode phone fax email 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 ??	0
counterexamples': [{u'table-content': [[u'employeeid', u'lastname', u'firstname', u'title', u'reportsto', u'birthdate', u'hiredate', u'address', u'city', u'state', u'country', u'postalcode', u'phone', u'fax', u'email'], [[[0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], 1]]], u'table-name': u'employee'}], u'rosette_result': u'NEQ'}	0
	0
	0
zufällig gleiches Ergebnis-Set aber andere Syntax & Semantik → Test für Verfahren mit pragmatischen Ansatz	0
	0

kein Support für LIMIT	1
kein Support für LIMIT	1
zufällig gleiches Ergebnis-Set aber andere Syntax & Semantik → kein Support für LIMIT	1
Two queries are not equivalent. Counter Examples: (i.e., input tables that, when fed into your input queries, will return different results) Table invoiceline invoicelineidinvoiceidtrackidunitpricequantity -10-101 01010 ??	0
kein Support für LIMIT	1
kein Support für float/numeric	1
kein Support für avg	1
	0
kein Support für LIKE	1
kein Support für LIKE	1
kein Support für LIKE	1
kein Support für LIKE	1
kein Support für IN	1

kein Support für IN	1
kein Support für between	1
kein Support für between	1
kein Support für between	1
	1
kein Support für UNION	1
kein Support für UNION	1
kein Support für UNION	1
	0

Two queries are not equivalent. Counter Examples: (i.e., input tables that, when fed into your input queries, will return different results) Table invoice invoiceidcustomeridinvoicedatebillingaddressbillingcitybillingstatebillingcountrybillingpostalcodetotal 000000000 000000000 000000000 000000000 000000000 000000000 000000000	0
See #50	0
	1
	0
kein Support für IN	1

internal error: out of time	1	
	1	
kein support für ORDER BY	1	
	0	
		%
Expected != actual:	35	61,4
error	34	59,6

Hinweise

Ergebnis (0: nicht äquivalent, 1: äquivalent, (pm=0..1): tw. Äquivalent, error: Fehler → System kann Querys nicht vergleichen)

pm = partial marking (z.B. zu 85% korrekt)

pm, sonst 0 → falls Verfahren partielle Bewertung unterstützt wird pm erwartet, ansonsten nicht äquivalent erwartet

Name = SQLAutoGrader, pm = false

				Ergebnis	
Mutations-Kategorie/ SQL-Feature	Nr.	Query (Musterlösung)	Test-Query (Kandidat)	erwartet	tats.
columns select vs. Select *	1	SELECT * FROM Album	SELECT AlbumId, Title, ArtistId FROM Album	1	0
	2	SELECT AlbumId, Title, ArtistId FROM Album	SELECT * FROM Album	1	0
where: same semantic, minimal syntax change	3	SELECT TrackId FROM PlaylistTrack WHERE TrackId < 100	SELECT TrackId FROM PlaylistTrack WHERE TrackId <= 99	1	1
Semicolon ;	4	SELECT Name FROM Genre;	SELECT Name FROM Genre	1	error
syntax sql keywords (upper, lower case)	5	SELECT Name FROM MediaType	select Name FrOm MediaType	1	1
column count differing	6	SELECT InvoiceDate, BillingAddress, BillingCity FROM Invoice	SELECT BillingAddress, BillingCity FROM Invoice	0	0
	7	SELECT BillingAddress, BillingCity FROM Invoice	SELECT InvoiceDate, BillingAddress, BillingCity FROM Invoice	0	1
column ordering	8		SELECT BillingCity, BillingAddress FROM Invoice	0	1
	9	SELECT TrackId, Name, GenreId from Track ORDER BY GenreId ASC	SELECT TrackId, Name, GenreId from Track ORDER BY GenreId	1	1

order by	10	SELECT TrackId, Name, GenreId from Track ORDER BY GenreId DESC	SELECT TrackId, Name, GenreId from Track	0	1
	11		SELECT TrackId, Name, GenreId from Track ORDER BY GenreId ASC	0	1
Redundanzen	12	SELECT Name, MediaTypeId FROM Track WHERE MediaTypeId > 2	SELECT Name, MediaTypeId FROM Track WHERE MediaTypeId > 2 and MediaTypeId > 1	1	1
	13		SELECT Name, MediaTypeId FROM Track WHERE MediaTypeId > 2 or MediaTypeId > 1	0	0
Sub-Query/Join	14	select Track.TrackId, Track.Name, Track.Composer from Track join (select MediaType.MediaTypeId, MediaType.Name from MediaType where MediaType.Name = 'AAC audio file') as t_MediaType ON Track.MediaTypeId = t_MediaType.MediaTypeId	select Track.TrackId, Track.Name, Track.Composer from Track join (select MediaType.MediaTypeId, MediaType.Name from MediaType) as t_MediaType ON Track.MediaTypeId = t_MediaType.MediaTypeId	0	0
	15		select Track.TrackId, Track.Name, Track.Composer from Track Join MediaType ON Track.MediaTypeId = MediaType.MediaTypeId	0	0
	16		select Track.TrackId, Track.Name, Track.Composer from Track join MediaType ON Track.MediaTypeId = MediaType.MediaTypeId WHERE MediaType.Name = 'AAC audio file'	1	0

	17		select Track.TrackId, Track.Name, Track.Composer from Track join MediaType ON Track.MediaTypeId = MediaType.MediaTypeId AND MediaType.Name = 'AAC audio file'	1	0
	18		select Track.TrackId, Track.Name, Track.Composer from Track, MediaType where Track.MediaTypeId = MediaType.MediaTypeId and (MediaType.Name = 'AAC audio file')	1	0
distinct	19	SELECT DISTINCT PlaylistId FROM PlaylistTrack	SELECT PlaylistId FROM PlaylistTrack	0	0

where, and	20	SELECT * FROM Employee
		WHERE Title = 'Sales Support Agent'
			SELECT * FROM Employee	0	0

	21		SELECT * FROM Employee
			WHERE ReportsTo = 2
				0	0

	22	SELECT Email FROM Employee
		WHERE City = 'Calgary' AND ReportsTo = 1
			SELECT Email FROM Employee
			WHERE City = 'Calgary'
				0	0

or	23	SELECT LastName, FirstName, City FROM Employee
		WHERE City = 'Edmonton' OR City = 'Lethbridge'
			SELECT LastName, FirstName, City FROM Employee
			WHERE City = 'Edmonton'
				0	0

	24		SELECT LastName, FirstName, City FROM Employee
			WHERE City = 'Edmonton' OR City = 'Calgary'
				0	0

not	25	SELECT LastName, FirstName, City FROM Employee
		WHERE NOT City = 'Calgary'
			SELECT LastName, FirstName, City FROM Employee
			WHERE City = 'Lethbridge' OR City = 'Edmonton'
				0	1

	26		SELECT LastName, FirstName, City FROM Employee
			WHERE NOT City = 'Edmonton'
				0	0

limit	27	SELECT LastName, FirstName, City FROM Employee LIMIT 3	SELECT LastName, FirstName, City FROM Employee LIMIT 4	0	0
	28		SELECT LastName, FirstName, City FROM Employee	0	0
min, max	29	SELECT MIN(UnitPrice) FROM InvoiceLine	SELECT UnitPrice FROM InvoiceLine LIMIT 1	0	0
	30		SELECT MAX(UnitPrice) FROM InvoiceLine	0	0

	31	SELECT MAX(UnitPrice) FROM InvoiceLine	SELECT UnitPrice FROM InvoiceLine LIMIT 1	0	0
count	32	SELECT COUNT(UnitPrice) FROM InvoiceLine WHERE UnitPrice = 0.99	SELECT UnitPrice FROM InvoiceLine WHERE UnitPrice = 0.99	0	0
avg	33	SELECT avg(Total) FROM Invoice	SELECT Total FROM Invoice	0	0
sum	34	SELECT sum(Total) FROM Invoice		0	0
like	35	SELECT ArtistId, Name FROM Artist WHERE Name LIKE '%Metal%'	SELECT ArtistId, Name FROM Artist	0	0
	36	SELECT ArtistId, Name FROM Artist WHERE Name LIKE 'Sant%'	SELECT ArtistId, Name FROM Artist WHERE Name LIKE 'San%'	0	0
	37	SELECT ArtistId, Name FROM Artist WHERE Name LIKE 'M%o'	SELECT ArtistId, Name FROM Artist WHERE Name LIKE 'M%'	0	0
	38	SELECT ArtistId, Name FROM Artist WHERE Name LIKE '_ %'	SELECT ArtistId, Name FROM Artist WHERE Name LIKE '_%'	0	0
in	39	SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice	SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice	0	error

	40	WHERE BillingCountry IN ('USA', 'Canada', 'Brazil')	SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice
			WHERE BillingCountry IN ('USA', 'Brazil')
				0	error

between	41	SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice
		WHERE InvoiceDate between '2012-06-04 00:00:00' AND '2012-07-28 00:00:00'
			SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice	0	0

	42		SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice
			WHERE InvoiceDate between '2012-05-04 00:00:00' AND '2012-07-28 00:00:00'
				0	0

	43	SELECT InvoiceId, CustomerId FROM Invoice
		WHERE InvoiceId between 50 AND 100
			SELECT InvoiceId, CustomerId FROM Invoice
			WHERE InvoiceId between 0 AND 120
				0	0

column alias	44	SELECT Name AS Genres FROM Genre	SELECT Name AS Genre FROM Genre	0	1

union	45	SELECT City FROM Customer
		UNION
		SELECT City FROM Employee
			SELECT City FROM Customer	0	0

	46		SELECT City FROM Customer
			UNION
			SELECT BillingCity FROM Invoice
				0	0

	47	SELECT City FROM Customer
		UNION ALL
		SELECT City FROM Employee
			SELECT City FROM Customer
			UNION
			SELECT City FROM Employee
				0	error

group by, having	49	SELECT InvoiceId, CustomerId, count(CustomerId) FROM Invoice GROUP BY CustomerId;	SELECT InvoiceId, CustomerId, count(CustomerId) FROM Invoice GROUP BY InvoiceId	0	0
	50	SELECT CustomerId, count(CustomerId) FROM Invoice group by CustomerId having count(CustomerId) < 7	SELECT CustomerId, count(CustomerId) FROM Invoice group by CustomerId	0	0
	51		SELECT CustomerId, count(CustomerId) FROM Invoice group by CustomerId having count(CustomerId) > 1	0	0

exists	52	SELECT TrackID, Name, GenreID
		FROM Track
		WHERE EXISTS
		(SELECT GenreID FROM Genre WHERE
		Track.GenreId = Genre.GenreId AND
			SELECT TrackID, Name, GenreID
			FROM Track
			WHERE EXISTS
			(SELECT GenreID FROM Genre WHERE Track.GenreId = Genre.GenreId AND Genre.Name = 'Reggae' OR Genre.Name = 'Jazz')
				0	error

	53		SELECT TrackID, Name, GenreID
			FROM Track
				0	error

	54	Track.GenreId = Genre.GenreId AND
		(Genre.Name = 'Reggae' OR Genre.Name = 'Jazz'))
			SELECT TrackID, Name, GenreID
			FROM Track
			WHERE GenreID IN
			(SELECT GenreID FROM Genre WHERE Track.GenreId = Genre.GenreId AND
			(Genre.Name = 'Reggae' OR Genre.Name = 'Jazz'))
				1	error

Inner, left, self join	55	SELECT PlaylistTrack.PlaylistId, Track.TrackId, Track.Name, Genre.Name FROM ((PlaylistTrack INNER JOIN Track ON PlaylistTrack.TrackId = Track.TrackId) INNER join Genre ON Genre.GenreId = Track.GenreId)	SELECT PlaylistTrack.PlaylistId, Track.TrackId, Track.Name, Genre.Name FROM ((PlaylistTrack JOIN Track ON PlaylistTrack.TrackId = Track.TrackId) INNER join Genre ON Track.GenreId = Genre.GenreId)	1	error

	56	SELECT Customer.CustomerId, Invoice.InvoiceId
		FROM Customer LEFT JOIN Invoice
		ON Customer.CustomerId = Invoice.InvoiceId
			SELECT Customer.CustomerId, Invoice.InvoiceId
			FROM Customer LEFT OUTER JOIN Invoice
			ON Customer.CustomerId = Invoice.InvoiceId
				1	error

	57	SELECT A.FirstName AS CustomerName1, B.FirstName AS CustomerName2, A.City
		FROM Customer A, Customer B
		WHERE A.CustomerID != B.CustomerID
		AND A.City = B.City
		ORDER BY A.City
			SELECT X.FirstName AS CustomerName1, Y.FirstName AS CustomerName2, Y.City
			FROM Customer X, Customer Y
			WHERE Y.CustomerID <> X.CustomerID
			AND X.City = Y.City
			ORDER BY Y.City
				1	0

to much joins	58	SELECT PlaylistTrack.PlaylistId, Track.TrackId, Track.Name FROM (PlaylistTrack INNER JOIN Track ON PlaylistTrack.TrackId = Track.TrackId)	SELECT PlaylistTrack.PlaylistId, Track.TrackId, Track.Name, Genre.Name FROM ((PlaylistTrack INNER JOIN Track ON PlaylistTrack.TrackId = Track.TrackId) INNER join Genre ON Genre.GenreId = Track.GenreId)	0	error

tats. = tatsächlich

Anmerkung	Compare
Output comparision → bug?!	1
Output rowcount Row count matched Expected:347 Obtained:347 Output comparision Output mismatch by rows	1
	0
(node:6661) UnhandledPromiseRejectionWarning: SyntaxError: Expected ",", ".", "AS", "GROUP", "INNER", "JOIN", "LEFT", "LIMIT", "ORDER", "UNION", "WHERE", "`", WHITE_SPACE, [A-Za-z0-9_], [A-Za-z_], or end of input but ";" found.	1
	0
erkennt untersch. Anzahl spalten direkt in SELECT nicht! → code geprüft: mysql.js:108; irreführende Fehlermeldung	0
tool erkennt untersch. Anzahl Spalten nicht als Fehler	1
	1
	0

	SQLAutoGrader-MySQL
	1
	1
	0
	0
Row count mismatched Expected:11 Obtained:3503 Missing Tables in query: rep_string_1 Extra Tables in query: rep_string_0	0
Output rowcount Row count mismatched Expected:11 Obtained:3503 Output comparision Output mismatch by 3492 rows Query Cluase Analysis Missing Tables in query: rep_string_1 Extra Tables in query: MediaType	0
Query eigentlich korrekt, nur laut Tool fehlen bestimmte Klauseln: Query Cluase Analysis Missing Tables in query: rep_string_1 Extra Tables in query: MediaType Extra condition on columns: Name	1

Query Cluase Analysis Missing Tables in query: rep_string_1 Extra Tables in query: MediaType	1
Query Cluase Analysis Missing Tables in query: rep_string_1 Extra Tables in query: MediaType Extra condition on columns: MediaTypeId,Name	1
	0
Query Output Analysis Output rowcount Row count mismatched Expected:3 Obtained:8 Output comparision Output mismatch by rows Query Cluase Analysis Missing condition on columns: Title	0
zufällig gleiches Ergebnis-Set aber andere Syntax & Semantik – Query Cluase Analysis Missing condition on columns: Title Extra condition on columns: ReportsTo	0
Query Output Analysis Output rowcount Row count mismatched Expected:2 Obtained:5 Output comparision Output mismatch by 3 rows Query Cluase Analysis Missing condition on columns: ReportsTo	0

Output rowcount Row count mismatched Expected:3 Obtained:1 Output comparision Output mismatch by 2 rows	0
Output rowcount Row count mismatched Expected:3 Obtained:6 Output comparision Output mismatch by 7 rows	0
zufällig gleiches Ergebnis-Set aber andere Syntax & Semantik → Test für Verfahren mit pragmatischen Ansatz	1
Output rowcount Row count mismatched Expected:3 Obtained:7 Output comparision Output mismatch by 6 rows	0
	0
	0
zufällig gleiches Ergebnis-Set aber andere Syntax & Semantik – Output rowcount Row count matched Expected:1 Obtained:1 Output comparision Output mismatch by rows	0
Query Output Analysis Output rowcount Row count matched Expected:1 Obtained:1 Output comparision Output mismatch by rows	0

Output rowcount Row count matched Expected:1 Obtained:1 Output comparision Output mismatch by rows	0
Output rowcount Row count mismatched Expected:1 Obtained:2129 Output comparision Output mismatch by rows	0
Output rowcount Row count mismatched Expected:1 Obtained:412 Output comparision Output mismatch by rows	0
	0
Output rowcount Row count mismatched Expected:1 Obtained:275 Output comparision Output mismatch by 274 rows Query Cluase Analysis Missing condition on columns: Name	0
row count mismatch	0
row count mismatch	0
row count mismatch	0
UnhandledPromiseRejectionWarning: SyntaxError: Expected "$", "(", or WHITE_SPACE but "r" found. at peg$buildStructuredError (/home/matthias/DEV/fsp-sqlequi/tools/SQLAutoGrader/node_modules/node-sqlparser/lib/parse.js:999:12)	1

UnhandledPromiseRejectionWarning: SyntaxError: Expected "$", "(", or WHITE_SPACE but "r" found. at peg$buildStructuredError (/home/matthias/DEV/fsp-sqlequi/tools/SQLAutoGrader/node_modules/node-sqlparser/lib/parse.js:999:12)	1
Output rowcount Row count mismatched Expected:13 Obtained:412 Output comparision Output mismatch by 399 rows Query Cluase Analysis Missing condition on columns: InvoiceDate,,	0
Query Output Analysis Output rowcount Row count mismatched Expected:13 Obtained:20 Output comparision Output mismatch by 7 rows Query Cluase Analysis Missing condition on columns: , Extra condition on columns: ,	0
irreführende Fehlermeldung siehe #42	0
Name der Spalte wird nicht berücksichtigt (Alias)	1
Output rowcount Row count mismatched Expected:55 Obtained:59 Output comparision Output mismatch by 2 rows	0
	0

UNION ALL crashes app	1
Output rowcount Row count matched Expected:412 Obtained:412 Output comparision Output mismatch by rows	0
Query Output Analysis Output rowcount Row count mismatched Expected:1 Obtained:59 Output comparision Output mismatch by rows Query Cluase Analysis Missing condition on columns: CustomerId	0
Fehlermeldung auch nicht ganz korrekt, Bedingung (having) ist da aber FALSCH: Output rowcount Row count mismatched Expected:1 Obtained:59 Output comparision Output mismatch by rows Query Cluase Analysis Missing condition on columns: CustomerId Extra condition on columns: CustomerId	0
internal tool error: SyntaxError: Expected "(" or WHITE_SPACE but "r" found.	1
internal tool error: SyntaxError: Expected "(" or WHITE_SPACE but "r" found.	1

SyntaxError: Expected "$", "(", or WHITE_SPACE but "r" found.	1
Failed to failed to extract SQL clauses due to error in parsing – Failed to compare output due to error in parsing	1
SyntaxError: Expected "JOIN" or WHITE_SPACE but "O" found.	1
Eigentlich korrekt, auch Ausgabe, ABER OC ROT: Output comparision Output mismatch by 24 rows (bug?!)	1
Failed to failed to extract SQL clauses due to error in parsing Hat Problem Kandidat-Q. Zu parsen	1
	%

Expected != actual:	22	38,6
error	10	17,5

Hinweise

Ergebnis (0: nicht äquivalent, 1: äquivalent, (pm=0..1): tw. Äquivalent, error: Fehler → System kann Querys nicht vergleichen)

pm = partial marking (z.B. zu 85% korrekt)

pm, sonst 0 → falls Verfahren partielle Bewertung unterstützt wird pm erwartet, ansonsten nicht äquivalent erwartet

Name = SQLAutoGrader, pm = false

				Ergebnis	
Mutations-Kategorie/ SQL-Feature	Nr.	Query (Musterlösung)	Test-Query (Kandidat)	erwartet	tats.
columns select vs. Select *	1	SELECT * FROM Album	SELECT AlbumId, Title, ArtistId FROM Album	1	0
	2	SELECT AlbumId, Title, ArtistId FROM Album	SELECT * FROM Album	1	0
where: same semantic, minimal syntax change	3	SELECT TrackId FROM PlaylistTrack WHERE TrackId < 100	SELECT TrackId FROM PlaylistTrack WHERE TrackId <= 99	1	1
Semicolon ;	4	SELECT Name FROM Genre;	SELECT Name FROM Genre	1	error
syntax sql keywords (upper, lower case)	5	SELECT Name FROM MediaType	select Name FrOm MediaType	1	1
column count differing	6	SELECT InvoiceDate, BillingAddress, BillingCity FROM Invoice	SELECT BillingAddress, BillingCity FROM Invoice	0	0
	7	SELECT BillingAddress, BillingCity FROM Invoice	SELECT InvoiceDate, BillingAddress, BillingCity FROM Invoice	0	1
column ordering	8		SELECT BillingCity, BillingAddress FROM Invoice	0	1
order by	9	SELECT TrackId, Name, GenreId from Track ORDER BY GenreId ASC	SELECT TrackId, Name, GenreId from Track ORDER BY GenreId	1	1
	10	SELECT TrackId, Name, GenreId from Track	SELECT TrackId, Name, GenreId from Track	0	1

	11	ORDER BY GenreId DESC	SELECT TrackId, Name, GenreId from Track ORDER BY GenreId ASC	0	1
Redundanzen	12	SELECT Name, MediaTypeId FROM Track WHERE MediaTypeId > 2	SELECT Name, MediaTypeId FROM Track WHERE MediaTypeId > 2 and MediaTypeId > 1	1	1
	13		SELECT Name, MediaTypeId FROM Track WHERE MediaTypeId > 2 or MediaTypeId > 1	0	0
	14		select Track.TrackId, Track.Name, Track.Composer from Track join (select MediaType.MediaTypeId, MediaType.Name from MediaType) as t_MediaType ON Track.MediaTypeId = t_MediaType.MediaTypeId	0	0
	15		select Track.TrackId, Track.Name, Track.Composer from Track Join MediaType ON Track.MediaTypeId = MediaType.MediaTypeId	0	0

Sub-Query/Join	16	select Track.TrackId, Track.Name, Track.Composer from Track join (select MediaType.MediaTypeId, MediaType.Name from MediaType where MediaType.Name = 'AAC audio file') as t_MediaType ON Track.MediaTypeId = t_MediaType.MediaTypeId	select Track.TrackId, Track.Name, Track.Composer from Track join MediaType ON Track.MediaTypeId = MediaType.MediaTypeId WHERE MediaType.Name = 'AAC audio file'	1	0
	17		select Track.TrackId, Track.Name, Track.Composer from Track join MediaType ON Track.MediaTypeId = MediaType.MediaTypeId AND MediaType.Name = 'AAC audio file'	1	0
	18		select Track.TrackId, Track.Name, Track.Composer from Track, MediaType where Track.MediaTypeId = MediaType.MediaTypeId and (MediaType.Name = 'AAC audio file')	1	0
distinct	19	SELECT DISTINCT PlaylistId FROM PlaylistTrack	SELECT PlaylistId FROM PlaylistTrack	0	0

where, and	20	SELECT * FROM Employee
		WHERE Title = 'Sales Support Agent'
			SELECT * FROM Employee	0	0

	21		SELECT * FROM Employee
			WHERE ReportsTo = 2
				0	0

	22	SELECT Email FROM Employee
		WHERE City = 'Calgary' AND ReportsTo = 1
			SELECT Email FROM Employee
			WHERE City = 'Calgary'
				0	0

or	23	SELECT LastName, FirstName, City FROM Employee
		WHERE City = 'Edmonton' OR City = 'Lethbridge'
			SELECT LastName, FirstName, City FROM Employee
			WHERE City = 'Edmonton'
				0	0

	24		SELECT LastName, FirstName, City FROM Employee
			WHERE City = 'Edmonton' OR City = 'Calgary'
				0	0

not	25	SELECT LastName, FirstName, City FROM Employee
		WHERE NOT City = 'Calgary'
			SELECT LastName, FirstName, City FROM Employee
			WHERE City = 'Lethbridge' OR City = 'Edmonton'
				0	1

	26		SELECT LastName, FirstName, City FROM Employee
			WHERE NOT City = 'Edmonton'
				0	0

limit	27	SELECT LastName, FirstName, City FROM Employee LIMIT 3	SELECT LastName, FirstName, City FROM Employee LIMIT 4	0	0
	28		SELECT LastName, FirstName, City FROM Employee	0	0
min, max	29	SELECT MIN(UnitPrice) FROM InvoiceLine	SELECT UnitPrice FROM InvoiceLine LIMIT 1	0	0
	30		SELECT MAX(UnitPrice) FROM InvoiceLine	0	0
	31	SELECT MAX(UnitPrice) FROM InvoiceLine	SELECT UnitPrice FROM InvoiceLine LIMIT 1	0	0

count	32	SELECT COUNT(UnitPrice) FROM InvoiceLine WHERE UnitPrice = 0.99	SELECT UnitPrice FROM InvoiceLine WHERE UnitPrice = 0.99	0	0
avg	33	SELECT avg(Total) FROM Invoice	SELECT Total FROM Invoice	0	0
sum	34	SELECT sum(Total) FROM Invoice		0	0
like	35	SELECT ArtistId, Name FROM Artist WHERE Name LIKE '%Metal%'	SELECT ArtistId, Name FROM Artist	0	0
	36	SELECT ArtistId, Name FROM Artist WHERE Name LIKE 'Sant%'	SELECT ArtistId, Name FROM Artist WHERE Name LIKE 'San%'	0	0
	37	SELECT ArtistId, Name FROM Artist WHERE Name LIKE 'M%o'	SELECT ArtistId, Name FROM Artist WHERE Name LIKE 'M%'	0	0
	38	SELECT ArtistId, Name FROM Artist WHERE Name LIKE '_ %'	SELECT ArtistId, Name FROM Artist WHERE Name LIKE '_%'	0	0
in	39	SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice WHERE BillingCountry IN ('USA', 'Canada', 'Brazil')	SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice	0	error
	40		SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice WHERE BillingCountry IN ('USA', 'Brazil')	0	error
	41	SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice	SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice	0	0

between	42	WHERE InvoiceDate between '2012-06-04 00:00:00' AND '2012-07-28 00:00:00'	SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice
			WHERE InvoiceDate between '2012-05-04 00:00:00' AND '2012-07-28 00:00:00'
				0	0

	43	SELECT InvoiceId, CustomerId FROM Invoice
		WHERE InvoiceId between 50 AND 100
			SELECT InvoiceId, CustomerId FROM Invoice
			WHERE InvoiceId between 0 AND 120
				0	0

column alias	44	SELECT Name AS Genres FROM Genre	SELECT Name AS Genre FROM Genre	0	1

union	45	SELECT City FROM Customer
		UNION
		SELECT City FROM Employee
			SELECT City FROM Customer	0	0

	46		SELECT City FROM Customer
			UNION
			SELECT BillingCity FROM Invoice
				0	0

	47	SELECT City FROM Customer
		UNION ALL
		SELECT City FROM Employee
			SELECT City FROM Customer
			UNION
			SELECT City FROM Employee
				0	error

group by, having	49	SELECT InvoiceId, CustomerId, count(CustomerId) FROM Invoice GROUP BY CustomerId;	SELECT InvoiceId, CustomerId, count(CustomerId) FROM Invoice
			GROUP BY InvoiceId
				0	0

	50	SELECT CustomerId, count(CustomerId) FROM Invoice group by CustomerId having count(CustomerId) < 7	SELECT CustomerId, count(CustomerId) FROM Invoice group by CustomerId	0	0

	51		SELECT CustomerId, count(CustomerId)
			FROM Invoice
			group by CustomerId
			having count(CustomerId) > 1
				0	0

exists	52	SELECT TrackID, Name, GenreID
		FROM Track
		WHERE EXISTS
		(SELECT GenreID FROM Genre WHERE Track.GenreId = Genre.GenreId AND (Genre.Name = 'Reggae' OR Genre.Name = 'Jazz'))
			SELECT TrackID, Name, GenreID
			FROM Track
			WHERE EXISTS
			(SELECT GenreID FROM Genre WHERE Track.GenreId = Genre.GenreId AND Genre.Name = 'Reggae' OR Genre.Name = 'Jazz')
				0	error

	53		SELECT TrackID, Name, GenreID
			FROM Track
				0	error

	54		SELECT TrackID, Name, GenreID
			FROM Track
			WHERE GenreID IN
			(SELECT GenreID FROM Genre WHERE Track.GenreId = Genre.GenreId AND (Genre.Name = 'Reggae' OR Genre.Name = 'Jazz'))
				1	error

Inner, left, self join	55	SELECT PlaylistTrack.PlaylistId, Track.TrackId, Track.Name, Genre.Name FROM ((PlaylistTrack INNER JOIN Track ON PlaylistTrack.TrackId = Track.TrackId) INNER join Genre ON Genre.GenreId = Track.GenreId)	SELECT PlaylistTrack.PlaylistId, Track.TrackId, Track.Name, Genre.Name FROM ((PlaylistTrack JOIN Track ON PlaylistTrack.TrackId = Track.TrackId) INNER join Genre ON Track.GenreId = Genre.GenreId)	1	error

	56	SELECT Customer.CustomerId, Invoice.InvoiceId
		FROM Customer LEFT JOIN Invoice
		ON Customer.CustomerId = Invoice.InvoiceId
			SELECT Customer.CustomerId, Invoice.InvoiceId
			FROM Customer LEFT OUTER JOIN Invoice
			ON Customer.CustomerId = Invoice.InvoiceId
				1	error

	57	SELECT A.FirstName AS CustomerName1,
		B.FirstName AS CustomerName2, A.City
		FROM Customer A, Customer B
		WHERE A.CustomerID != B.CustomerID
		AND A.City = B.City
		ORDER BY A.City
			SELECT X.FirstName AS CustomerName1,
			Y.FirstName AS CustomerName2, Y.City
			FROM Customer X, Customer Y
			WHERE Y.CustomerID <> X.CustomerID
			AND X.City = Y.City
			ORDER BY Y.City
				1	0

to much joins	58	SELECT PlaylistTrack.PlaylistId, Track.TrackId, Track.Name FROM (PlaylistTrack INNER JOIN Track ON PlaylistTrack.TrackId = Track.TrackId)	SELECT PlaylistTrack.PlaylistId, Track.TrackId, Track.Name, Genre.Name FROM ((PlaylistTrack INNER JOIN Track ON PlaylistTrack.TrackId = Track.TrackId) INNER join Genre ON Genre.GenreId = Track.GenreId)	0	error

tats. = tatsächlich

Anmerkung	Compare	MySQL	PostgreSQL
siehe MySQL	1	0	0
siehe MySQL	1	0	0
	0	1	1
UnhandledPromiseRejectionWarning: SyntaxError: Expected ",", ".", "AS", "GROUP", "INNER", "JOIN", "LEFT", "LIMIT", "ORDER", "UNION", "WHERE", "`", WHITE_SPACE, [A-Za-z0-9_], [A-Za-z_], or end of input but ";" found.	1	error	error
	0	1	1
Output rowcount Row count matched Expected:412 Obtained:412 Output comparision Output mismatch by rows	0	0	0
	1	1	1
	1	1	1
	0	1	1
	1	1	1

	1	1	1
	0	1	1
Output rowcount Row count mismatched Expected:232 Obtained:469 Output comparision Output mismatch by 237 rows	0	0	0
Query Output Analysis Output rowcount Row count mismatched Expected:11 Obtained:3503 Output comparision Output mismatch by 3492 rows Query Cluase Analysis Missing Tables in query: rep_string_1 Extra Tables in query: rep_string_0	0	0	0
Output rowcount Row count mismatched Expected:11 Obtained:3503 Output comparision Output mismatch by 3492 rows Query Cluase Analysis Missing Tables in query: rep_string_1 Extra Tables in query: MediaType	0	0	0

Output rowcount Row count matched Expected:11 Obtained:11 Output comparision Output matches exactly Query Cluase Analysis Missing Tables in query: rep_string_1 Extra Tables in query: MediaType Extra condition on columns: Name	1	0	0
Missing Tables in query: rep_string_1 Extra Tables in query: MediaType	1	0	0
Missing Tables in query: rep_string_1 Extra Tables in query: MediaType Extra condition on columns: MediaTypeId,Name	1	0	0
Row count mismatched Expected:14 Obtained:8715	0	0	0
	0	0	0
zufällig gleiches Ergebnis-Set aber andere Syntax & Semantik – Output comparision Output mismatch by rows Query Cluase Analysis Missing condition on columns: Title Extra condition on columns: ReportsTo	0	0	0

Output rowcount Row count mismatched Expected:2 Obtained:5 Output comparision Output mismatch by 3 rows Query Cluase Analysis Missing condition on columns: ReportsTo	0	0	0
Output rowcount Row count mismatched Expected:3 Obtained:1 Output comparision Output mismatch by 2 rows	0	0	0
Output rowcount Row count mismatched Expected:3 Obtained:6 Output comparision Output mismatch by 7 rows	0	0	0
zufällig gleiches Ergebnis-Set aber andere Syntax & Semantik → Test für Verfahren mit pragmatischen Ansatz	1	1	1
	0	0	0
Output rowcount Row count mismatched Expected:3 Obtained:4 Output comparision Output mismatch by 1 rows	0	0	0
	0	0	0
zufällig gleiches Ergebnis-Set aber andere Syntax & Semantik → „Output comparision Output mismatch by rows“	0	0	0
„Output mismatch by rows“	0	0	0
„Output mismatch by rows“	0	0	0

	0	0	0
	0	0	0
	0	0	0
Output rowcount Row count mismatched Expected:1 Obtained:275 Output comparision Output mismatch by 274 rows Query Cluase Analysis Missing condition on columns: Name	0	0	0
Output rowcount Row count mismatched Expected:9 Obtained:10 Output comparision Output mismatch by 1 rows	0	0	0
Output rowcount Row count mismatched Expected:3 Obtained:20 Output comparision Output mismatch by 17 rows	0	0	0
Output rowcount Row count mismatched Expected:3 Obtained:275 Output comparision Output mismatch by 272 rows	0	0	0
: SyntaxError: Expected "$", "(", or WHITE_SPACE but "r" found.	1	error	error
: SyntaxError: Expected "$", "(", or WHITE_SPACE but "r" found.	1	error	error
Output rowcount Row count mismatched Expected:13 Obtained:412 Output comparision Output mismatch by 399 rows	0	0	0

Output rowcount Row count mismatched Expected:13 Obtained:20 Output comparision Output mismatch by 7 rows	0	0	0
Output rowcount Row count mismatched Expected:51 Obtained:120 Output comparision Output mismatch by 69 rows	0	0	0
	1	1	1
Output rowcount Row count mismatched Expected:55 Obtained:59 Output comparision Output mismatch by 2 rows	0	0	0
Output rowcount Row count mismatched Expected:55 Obtained:53 Output comparision Output mismatch by 2 rows	0	0	0
UNION ALL crashes app: SyntaxError: Expected "(", "SELECT", or WHITE_SPACE but "A" found.	1	error	error
Output rowcount Row count matched Expected:412 Obtained:412 Output comparision Output mismatch by rows	0		
Output rowcount Row count mismatched Expected:1 Obtained:59 Output comparision Output mismatch by rows – Missing condition on columns: CustomerId	0	0	0

Output rowcount Row count mismatched Expected:1 Obtained:59 Output comparision Output mismatch by rows Query Cluase Analysis Missing condition on columns: CustomerId Extra condition on columns: CustomerId	0	0	0
SyntaxError: Expected "(" or WHITE_SPACE but "r" found.	1	error	error
SyntaxError: Expected "(" or WHITE_SPACE but "r" found.	1	error	error
SyntaxError: Expected "(" or WHITE_SPACE but "r" found.	1	error	error
Failed to failed to extract SQL clauses due to error in parsing Query Output Analysis Output rowcount Row count matched Expected:8715 Obtained:8715 Output comparision Failed to compare output due to error in parsing	1	error	error
SyntaxError: Expected "JOIN" or WHITE_SPACE but "O" found.	1	error	error

Output rowcount Row count matched Expected:12 Obtained:12 Output comparision Output mismatch by 24 rows	1		0	0
Failed to compare output due to error in parsing	1		error	error
		%		
Expected != actual:	22	38,6	diff=	0
error	10	17,5		

diff
0
0
0
0
0
0
0
0
0
0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0
0

0

0

0

0

0

0

0

0

0
0

0

0

0

0

0

0

0

0

0

0

0

0

Hinweise

Ergebnis (0: nicht äquivalent, 1: äquivalent, (pm=0..1): tw. Äquivalent, error: Fehler → System kann Querys nicht vergleichen)

pm = partial marking (z.B. zu 85% korrekt)

pm → falls Verfahren partielle Bewertung unterstützt wird pm erwartet, ansonsten nicht äquivalent erwartet

Name = XData, pm = true

				Ergebnis	
Mutations-Kategorie/ SQL-Feature	Nr.	Query (Musterlösung)	Test-Query (Kandidat)	erwartet	tats.
columns select vs. Select *	1	SELECT * FROM Album;	SELECT AlbumId, Title, ArtistId FROM Album;	1	1
	2	SELECT AlbumId, Title, ArtistId FROM Album;	SELECT * FROM Album;	1	1
where: same semantic, minimal syntax change	3	SELECT TrackId FROM PlaylistTrack WHERE TrackId < 100;	SELECT TrackId FROM PlaylistTrack WHERE TrackId <= 99;	1	1
Semicolon ;	4	SELECT Name FROM Genre;	SELECT Name FROM Genre	1	1
syntax sql keywords (upper, lower case)	5	SELECT Name FROM MediaType;	select Name FrOm MediaType;	1	1
column count differing	6	SELECT InvoiceDate, BillingAddress, BillingCity FROM Invoice;	SELECT BillingAddress, BillingCity FROM Invoice;	pm	0,75
	7	SELECT BillingAddress, BillingCity FROM Invoice;	SELECT InvoiceDate, BillingAddress, BillingCity FROM Invoice;	pm	0,833
column ordering	8		SELECT BillingCity, BillingAddress FROM Invoice;	pm	1
order by	9	SELECT TrackId, Name, GenreId from Track ORDER BY GenreId ASC;	SELECT TrackId, Name, GenreId from Track ORDER BY GenreId;	1	1
	10	SELECT TrackId, Name, GenreId from Track ORDER BY GenreId DESC;	SELECT TrackId, Name, GenreId from Track;	0	1
	11		SELECT TrackId, Name, GenreId from Track ORDER BY GenreId ASC;	pm	1
	12	SELECT Name, MediaTypeId FROM Track	SELECT Name, MediaTypeId FROM Track WHERE MediaTypeId > 2 and MediaTypeId > 1;	1	1

Redundanzen	13	WHERE MediaTypeId > 2;	SELECT Name, MediaTypeId FROM Track WHERE MediaTypeId > 2 or MediaTypeId > 1;	pm	0,167
Sub-Query/Join	14	select Track.TrackId, Track.Name, Track.Composer from Track join (select MediaType.MediaTypeId, MediaType.Name from MediaType where MediaType.Name = 'AAC audio file') as t_MediaType ON Track.MediaTypeId = t_MediaType.MediaTypeId;	select Track.TrackId, Track.Name, Track.Composer from Track join (select MediaType.MediaTypeId, MediaType.Name from MediaType) as t_MediaType ON Track.MediaTypeId = t_MediaType.MediaTypeId;	pm	0,727
	15		select Track.TrackId, Track.Name, Track.Composer from Track Join MediaType ON Track.MediaTypeId = MediaType.MediaTypeId;	pm	0,727
	16		select Track.TrackId, Track.Name, Track.Composer from Track join MediaType ON Track.MediaTypeId = MediaType.MediaTypeId WHERE MediaType.Name = 'AAC audio file';	1	1
	17		select Track.TrackId, Track.Name, Track.Composer from Track join MediaType ON Track.MediaTypeId = MediaType.MediaTypeId AND MediaType.Name = 'AAC audio file';	1	1

	18		select Track.TrackId, Track.Name, Track.Composer from Track, MediaType where Track.MediaTypeId = MediaType.MediaTypeId and (MediaType.Name = 'AAC audio file')	1	1
distinct	19	SELECT DISTINCT PlaylistId FROM PlaylistTrack;	SELECT PlaylistId FROM PlaylistTrack;	0	0,667

where, and	20	SELECT * FROM Employee
		WHERE Title = 'Sales Support Agent';
			SELECT * FROM Employee;	0	0,842

	21		SELECT * FROM Employee
			WHERE ReportsTo = 2;
				pm	1

	22	SELECT Email FROM Employee
		WHERE City = 'Calgary' AND ReportsTo = 1;
			SELECT Email FROM Employee
			WHERE City = 'Calgary';
				pm	0,625

or	23	SELECT LastName, FirstName, City FROM Employee
		WHERE City = 'Edmonton' OR City = 'Lethbridge';
			SELECT LastName, FirstName, City FROM Employee
			WHERE City = 'Edmonton';
				pm	0,7

	24		SELECT LastName, FirstName, City FROM Employee
			WHERE City = 'Edmonton' OR City = 'Calgary';
				pm	0,9

not	25	SELECT LastName, FirstName, City FROM Employee
		WHERE NOT City = 'Calgary';
			SELECT LastName, FirstName, City FROM Employee
			WHERE City = 'Lethbridge' OR City = 'Edmonton';
				pm	1

	26		SELECT LastName, FirstName, City FROM Employee
			WHERE NOT City = 'Edmonton';
				pm	0,857

limit	27	SELECT LastName, FirstName, City FROM Employee
		LIMIT 3;
			SELECT LastName, FirstName, City FROM Employee
			LIMIT 4;
				pm	1

	28		SELECT LastName, FirstName, City FROM Employee;	0	1
	29	SELECT MIN(UnitPrice) FROM InvoiceLine;	SELECT UnitPrice FROM InvoiceLine LIMIT 1;	0	1

min, max	30	SELECT MIN(UnitPrice) FROM InvoiceLine;	SELECT MAX(UnitPrice) FROM InvoiceLine;	pm	0,5
	31	SELECT MAX(UnitPrice) FROM InvoiceLine;	SELECT UnitPrice FROM InvoiceLine LIMIT 1;	0	0,5
count	32	SELECT COUNT(UnitPrice) FROM InvoiceLine WHERE UnitPrice = 0.99;	SELECT UnitPrice FROM InvoiceLine WHERE UnitPrice = 0.99;	0	0,8
avg	33	SELECT avg(Total) FROM Invoice;	SELECT Total FROM Invoice;	0	0,5
sum	34	SELECT sum(Total) FROM Invoice;		0	0,5
like	35	SELECT ArtistId, Name FROM Artist WHERE Name LIKE '%Metal%';	SELECT ArtistId, Name FROM Artist;	0	0,5
	36	SELECT ArtistId, Name FROM Artist WHERE Name LIKE 'Sant%';	SELECT ArtistId, Name FROM Artist WHERE Name LIKE 'San%';	pm	0,833
	37	SELECT ArtistId, Name FROM Artist WHERE Name LIKE 'M%o';	SELECT ArtistId, Name FROM Artist WHERE Name LIKE 'M%';	pm	0,833
	38	SELECT ArtistId, Name FROM Artist WHERE Name LIKE '_ %';	SELECT ArtistId, Name FROM Artist WHERE Name LIKE '_%';	pm	0,833
	40		SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice WHERE BillingCountry IN ('USA', 'Brazil');	pm	1
between	41	SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice WHERE InvoiceDate between '2012-06-04 00:00:00' AND '2012-07-28 00:00:00';	SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice;	0	0,4
	42		SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice WHERE InvoiceDate between '2012-05-04 00:00:00' AND '2012-07-28 00:00:00';	pm	0,9
	43	SELECT InvoiceId, CustomerId FROM Invoice WHERE InvoiceId between 50 AND 100;	SELECT InvoiceId, CustomerId FROM Invoice WHERE InvoiceId between 0 AND 120;	pm	0,778
column alias	44	SELECT Name AS Genres FROM Genre;	SELECT Name AS Genre FROM Genre;	pm	1
union	45	SELECT City FROM Customer UNION SELECT City FROM Employee;	SELECT City FROM Customer;	0	0,4
	46		SELECT City FROM Customer UNION SELECT BillingCity FROM Invoice;	pm	0,6

union	47	SELECT City FROM Customer
		UNION ALL
		SELECT City FROM Employee;
			SELECT City FROM Customer
			UNION
			SELECT City FROM Employee;
				pm	1

group by, having	49	SELECT InvoiceId, CustomerId, count(CustomerId) FROM Invoice GROUP BY CustomerId;	SELECT InvoiceId, CustomerId, count(CustomerId) FROM Invoice GROUP BY InvoiceId;	pm	1

	50	SELECT CustomerId, count(CustomerId) FROM Invoice
		group by CustomerId
		having count(CustomerId) < 7;
			SELECT CustomerId, count(CustomerId)
			FROM Invoice
			group by CustomerId;
				0	1

	51		SELECT CustomerId, count(CustomerId)
			FROM Invoice
			group by CustomerId
			having count(CustomerId) > 1;
				pm	1

exists	52	SELECT TrackID, Name, GenreID
		FROM Track
		WHERE EXISTS
		(SELECT GenreID FROM Genre WHERE Track.GenreId = Genre.GenreId AND (Genre.Name = 'Reggae' OR Genre.Name = 'Jazz'));
			SELECT TrackID, Name, GenreID
			FROM Track
			WHERE EXISTS
			(SELECT GenreID FROM Genre WHERE Track.GenreId = Genre.GenreId AND Genre.Name = 'Reggae' OR Genre.Name = 'Jazz');
				pm	1

	53		SELECT TrackID, Name, GenreID
			FROM Track;
				0	0,267

	54		SELECT TrackID, Name, GenreID
			FROM Track
			WHERE GenreID IN
			(SELECT GenreID FROM Genre WHERE Track.GenreId = Genre.GenreId AND (Genre.Name = 'Reggae' OR Genre.Name = 'Jazz'));
				1	1

	55	SELECT PlaylistTrack.PlaylistId, Track.TrackId, Track.Name, Genre.Name FROM
		((PlaylistTrack INNER JOIN Track ON PlaylistTrack.TrackId = Track.TrackId)
		INNER join
		Genre ON Genre.GenreId = Track.GenreId
		);
			SELECT PlaylistTrack.PlaylistId, Track.TrackId, Track.Name, Genre.Name
			FROM
			((PlaylistTrack JOIN Track ON PlaylistTrack.TrackId = Track.TrackId)
			INNER join
			Genre ON Track.GenreId = Genre.GenreId
			);
				1	1

Inner, left, self join	56	SELECT Customer.CustomerId, Invoice.InvoiceId
		FROM Customer LEFT JOIN Invoice
		ON Customer.CustomerId = Invoice.InvoiceId
			SELECT Customer.CustomerId, Invoice.InvoiceId
			FROM Customer LEFT OUTER JOIN Invoice
			ON Customer.CustomerId = Invoice.InvoiceId
				1	1

	57	SELECT A.FirstName AS CustomerName1, B.FirstName AS CustomerName2, A.City
		FROM Customer A, Customer B
		WHERE A.CustomerID != B.CustomerID
		AND A.City = B.City
		ORDER BY A.City;
			SELECT X.FirstName AS CustomerName1, Y.FirstName AS CustomerName2, Y.City
			FROM Customer X, Customer Y
			WHERE Y.CustomerID <> X.CustomerID
			AND X.City = Y.City
			ORDER BY Y.City;
				1	1

to much joins	58	SELECT PlaylistTrack.PlaylistId, Track.TrackId, Track.Name FROM
		(PlaylistTrack INNER JOIN Track ON PlaylistTrack.TrackId = Track.TrackId
		);
			SELECT PlaylistTrack.PlaylistId, Track.TrackId, Track.Name, Genre.Name FROM
			((PlaylistTrack INNER JOIN Track ON PlaylistTrack.TrackId = Track.TrackId)
			INNER join
			Genre ON Genre.GenreId = Track.GenreId
			);
				0	0

tats. = tatsächlich

Anmerkung	Compare
	0
	0
empty generated dataset options	0
	0
	0
empty generated dataset options	0
empty generated dataset options	0
	1
empty generated dataset options	0
empty generated dataset options	1
empty generated dataset options	1
	0

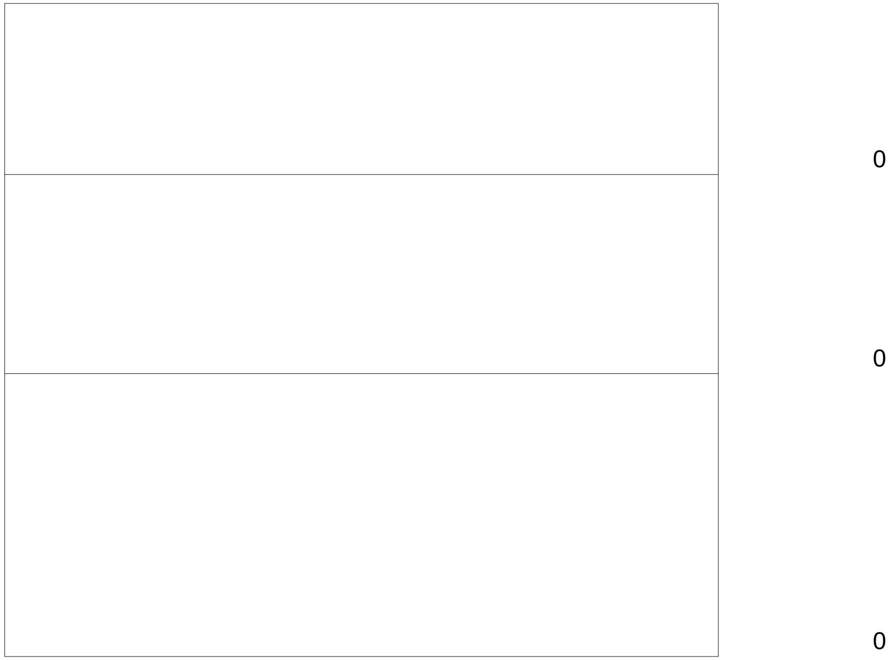
	XData
	0
	0
	0
	0
	0

	XData
	0
	1
	1
	1
	0
	0
	0
zufällig gleiches Ergebnis-Set aber andere Syntax & Semantik → Test für Verfahren mit pragmatischen Ansatz	1
	0
#fehler in auswertung des tools	1
#fehler in auswertung des tools	1
zufällig gleiches Ergebnis-Set aber andere Syntax & Semantik → Test für Verfahren mit pragmatischen Ansatz	1

	XData
#fehler in auswertung des tools	0
#fehler in auswertung des tools	1
#fehler in auswertung des tools	1
#fehler in auswertung des tools	1
#fehler in auswertung des tools	1
	1
	0
	0
	0
	1
	1
	0
good example of PM evaluation	0
	1
	1
	0

	XData
	1
	1
	1
	1
	1
	1
	0
	0

XData



		%
Expected != actual:	25	44,6
error	0	0

Hinweise

Ergebnis (0: nicht äquivalent, 1: äquivalent, (pm=0..1): tw. Äquivalent, error: Fehler → System kann Querys nicht vergleichen)

pm = partial marking (z.B. zu 85% korrekt)

pm → falls Verfahren partielle Bewertung unterstützt wird pm erwartet, ansonsten nicht äquivalent erwartet

Name = CS12x, pm = true

				Ergebnis	
Mutations-Kategorie/ SQL-Feature	Nr.	Query (Musterlösung)	Test-Query (Kandidat)	erwartet	tats.
columns select vs. Select *	1	SELECT * FROM Album;	SELECT AlbumId, Title, ArtistId FROM Album;	1	1
	2	SELECT AlbumId, Title, ArtistId FROM Album;	SELECT * FROM Album;	1	1
where: same semantic, minimal syntax change	3	SELECT TrackId FROM PlaylistTrack WHERE TrackId < 100;	SELECT TrackId FROM PlaylistTrack WHERE TrackId <= 99;	1	0
Semicolon ;	4	SELECT Name FROM Genre;	SELECT Name FROM Genre	1	1
syntax sql keywords (upper, lower case)	5	SELECT Name FROM MediaType;	select Name FrOm MediaType;	1	1
column count differing	6	SELECT InvoiceDate, BillingAddress, BillingCity FROM Invoice;	SELECT BillingAddress, BillingCity FROM Invoice;	pm	0
	7	SELECT BillingAddress, BillingCity FROM Invoice;	SELECT InvoiceDate, BillingAddress, BillingCity FROM Invoice;	pm	0
column ordering	8		SELECT BillingCity, BillingAddress FROM Invoice;	pm	0,7
order by	9	SELECT TrackId, Name, GenreId from Track ORDER BY GenreId ASC;	SELECT TrackId, Name, GenreId from Track ORDER BY GenreId;	1	1
	10	SELECT TrackId, Name, GenreId from Track ORDER BY GenreId DESC;	SELECT TrackId, Name, GenreId from Track;	0	0,7
	11		SELECT TrackId, Name, GenreId from Track ORDER BY GenreId ASC;	pm	0,7
	12	SELECT Name, MediaTypeId FROM Track	SELECT Name, MediaTypeId FROM Track WHERE MediaTypeId > 2 and MediaTypeId > 1;	1	1

Redundanzen	13	WHERE MediaTypeId > 2;	SELECT Name, MediaTypeId FROM Track WHERE MediaTypeId > 2 or MediaTypeId > 1;	pm	0
Sub-Query/Join	14	select Track.TrackId, Track.Name, Track.Composer from Track join (select MediaType.MediaTypeId, MediaType.Name from MediaType where MediaType.Name = 'AAC audio file') as t_MediaType ON Track.MediaTypeId = t_MediaType.MediaTypeId;	select Track.TrackId, Track.Name, Track.Composer from Track join (select MediaType.MediaTypeId, MediaType.Name from MediaType) as t_MediaType ON Track.MediaTypeId = t_MediaType.MediaTypeId;	pm	0
	15		select Track.TrackId, Track.Name, Track.Composer from Track Join MediaType ON Track.MediaTypeId = MediaType.MediaTypeId;	pm	0
	16		select Track.TrackId, Track.Name, Track.Composer from Track join MediaType ON Track.MediaTypeId = MediaType.MediaTypeId WHERE MediaType.Name = 'AAC audio file';	1	1
	17		select Track.TrackId, Track.Name, Track.Composer from Track join MediaType ON Track.MediaTypeId = MediaType.MediaTypeId AND MediaType.Name = 'AAC audio file';	1	1

	18		select Track.TrackId, Track.Name, Track.Composer from Track, MediaType where Track.MediaTypeId = MediaType.MediaTypeId and (MediaType.Name = 'AAC audio file')	1	1
distinct	19	SELECT DISTINCT PlaylistId FROM PlaylistTrack;	SELECT PlaylistId FROM PlaylistTrack;	0	0

where, and	20	SELECT * FROM Employee
		WHERE Title = 'Sales Support Agent';
			SELECT * FROM Employee;	0	0

	21		SELECT * FROM Employee
			WHERE ReportsTo = 2;
				pm	1

	22	SELECT Email FROM Employee
		WHERE City = 'Calgary' AND ReportsTo = 1;
			SELECT Email FROM Employee
			WHERE City = 'Calgary';
				pm	0

or	23	SELECT LastName, FirstName, City FROM Employee
		WHERE City = 'Edmonton' OR City = 'Lethbridge';
			SELECT LastName, FirstName, City FROM Employee
			WHERE City = 'Edmonton';
				pm	0

	24		SELECT LastName, FirstName, City FROM Employee
			WHERE City = 'Edmonton' OR City = 'Calgary';
				pm	0

not	25	SELECT LastName, FirstName, City FROM Employee
		WHERE NOT City = 'Calgary';
			SELECT LastName, FirstName, City FROM Employee
			WHERE City = 'Lethbridge' OR City = 'Edmonton';
				pm	1

	26		SELECT LastName, FirstName, City FROM Employee
			WHERE NOT City = 'Edmonton';
				pm	0

limit	27	SELECT LastName, FirstName, City FROM Employee
		LIMIT 3;
			SELECT LastName, FirstName, City FROM Employee
			LIMIT 4;
				pm	0

	28		SELECT LastName, FirstName, City FROM Employee;	0	0
	29	SELECT MIN(UnitPrice) FROM InvoiceLine;	SELECT UnitPrice FROM InvoiceLine LIMIT 1;	0	1

min, max	30	SELECT MIN(UnitPrice) FROM InvoiceLine;	SELECT MAX(UnitPrice) FROM InvoiceLine;	pm	0
	31	SELECT MAX(UnitPrice) FROM InvoiceLine;	SELECT UnitPrice FROM InvoiceLine LIMIT 1;	0	0
count	32	SELECT COUNT(UnitPrice) FROM InvoiceLine WHERE UnitPrice = 0.99;	SELECT UnitPrice FROM InvoiceLine WHERE UnitPrice = 0.99;	0	0
avg	33	SELECT avg(Total) FROM Invoice;	SELECT Total FROM Invoice;	0	0
sum	34	SELECT sum(Total) FROM Invoice;		0	0
like	35	SELECT ArtistId, Name FROM Artist WHERE Name LIKE '%Metal%';	SELECT ArtistId, Name FROM Artist;	0	0
	36	SELECT ArtistId, Name FROM Artist WHERE Name LIKE 'Sant%';	SELECT ArtistId, Name FROM Artist WHERE Name LIKE 'San%';	pm	0
	37	SELECT ArtistId, Name FROM Artist WHERE Name LIKE 'M%o';	SELECT ArtistId, Name FROM Artist WHERE Name LIKE 'M%';	pm	0
	38	SELECT ArtistId, Name FROM Artist WHERE Name LIKE '_ %';	SELECT ArtistId, Name FROM Artist WHERE Name LIKE '_%';	pm	0
in	39	SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice WHERE BillingCountry IN ('USA', 'Canada', 'Brazil');	SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice;	0	0
	40		SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice WHERE BillingCountry IN ('USA', 'Brazil');	pm	0
between	41	SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice WHERE InvoiceDate between '2012-06-04 00:00:00' AND '2012-07-28 00:00:00';	SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice;	0	0
	42		SELECT InvoiceId, InvoiceDate, BillingCountry FROM Invoice WHERE InvoiceDate between '2012-05-04 00:00:00' AND '2012-07-28 00:00:00';	pm	0
	43	SELECT InvoiceId, CustomerId FROM Invoice WHERE InvoiceId between 50 AND 100;	SELECT InvoiceId, CustomerId FROM Invoice WHERE InvoiceId between 0 AND 120;	pm	0
column alias	44	SELECT Name AS Genres FROM Genre;	SELECT Name AS Genre FROM Genre;	pm	1
	45	SELECT City FROM Customer	SELECT City FROM Customer;	0	0

union	46	SELECT City FROM Customer
		UNION
		SELECT City FROM Employee;
			SELECT City FROM Customer
			UNION
			SELECT BillingCity FROM Invoice;
				pm	0

	47	SELECT City FROM Customer
		UNION ALL
		SELECT City FROM Employee;
			SELECT City FROM Customer
			UNION
			SELECT City FROM Employee;
				pm	0

group by, having	49		**SELECT** InvoiceId, CustomerId,
			count(CustomerId) **FROM**
			Invoice**GROUP BY** InvoiceId;
				pm	0,7

	50	SELECT CustomerId, count(CustomerId) FROM Invoice
		group by CustomerId
		having count(CustomerId) < 7;
			SELECT CustomerId, count(CustomerId)
			FROM Invoice
			group by CustomerId;
				0	0

	51		SELECT CustomerId, count(CustomerId)
			FROM Invoice
			group by CustomerId
			having count(CustomerId) > 1;
				pm	0

exists	52	SELECT TrackID, Name, GenreID
		FROM Track
		WHERE EXISTS
		(SELECT GenreID FROM Genre WHERE Track.GenreId = Genre.GenreId AND (Genre.Name = 'Reggae' OR Genre.Name = 'Jazz'));
			SELECT TrackID, Name, GenreID
			FROM Track
			WHERE EXISTS
			(SELECT GenreID FROM Genre WHERE Track.GenreId = Genre.GenreId AND Genre.Name = 'Reggae' OR Genre.Name = 'Jazz');
				pm	0

	53		SELECT TrackID, Name, GenreID
			FROM Track;
				0	0

	54		SELECT TrackID, Name, GenreID
			FROM Track
			WHERE GenreID IN
			(SELECT GenreID FROM Genre WHERE Track.GenreId = Genre.GenreId AND (Genre.Name = 'Reggae' OR Genre.Name = 'Jazz'));
				1	0,7

Inner, left, self join	55	SELECT PlaylistTrack.PlaylistId, Track.TrackId, Track.Name, Genre.Name FROM ((PlaylistTrack INNER JOIN Track ON PlaylistTrack.TrackId = Track.TrackId) INNER join Genre ON Genre.GenreId = Track.GenreId);	SELECT PlaylistTrack.PlaylistId, Track.TrackId, Track.Name, Genre.Name FROM ((PlaylistTrack JOIN Track ON PlaylistTrack.TrackId = Track.TrackId) INNER join Genre ON Track.GenreId = Genre.GenreId);	1	1

	56	SELECT Customer.CustomerId, Invoice.InvoiceId
		FROM Customer LEFT JOIN Invoice
		ON Customer.CustomerId = Invoice.InvoiceId
			SELECT Customer.CustomerId,
			Invoice.InvoiceId
			FROM Customer LEFT OUTER JOIN
			Invoice
			ON Customer.CustomerId =
			Invoice.InvoiceId
				1	1

	57	SELECT A.FirstName AS CustomerName1,
		B.FirstName AS CustomerName2, A.City
		FROM Customer A, Customer B
		WHERE A.CustomerID != B.CustomerID
		AND A.City = B.City
		ORDER BY A.City;
			SELECT X.FirstName AS CustomerName1,
			Y.FirstName AS CustomerName2, Y.City
			FROM Customer X, Customer Y
			WHERE Y.CustomerID <> X.CustomerID
			AND X.City = Y.City
			ORDER BY Y.City;
				1	1

to much joins	58	SELECT PlaylistTrack.PlaylistId, Track.TrackId, Track.Name FROM (PlaylistTrack INNER JOIN Track ON PlaylistTrack.TrackId = Track.TrackId);	SELECT PlaylistTrack.PlaylistId, Track.TrackId, Track.Name, Genre.Name FROM ((PlaylistTrack INNER JOIN Track ON PlaylistTrack.TrackId = Track.TrackId) INNER join Genre ON Genre.GenreId = Track.GenreId);	0	0

tats. = tatsächlich

Anmerkung	Compare
	0
	0
BadQueryError [-0]: Query might be bad because it is empty, contains unexpected SQL or extra stuff before or after. This might need to be manually graded.	1
	0
	0
WrongNumColumnsError [-0]: More or fewer columns included.	1
WrongNumColumnsError [-0]: More or fewer columns included.	1
ColumnOrderError [-3]: Columns are in the wrong order.	0
	0
OrderByError [-3]: Missing or incorrect ORDER BY statement.	1
OrderByError [-3]: Missing or incorrect ORDER BY statement.	0
	0

	CS12x
differing row contents	1
differing row contents	1
differing row contents	1
	0
	0

	CS12x
	0
	0
	0
zufällig gleiches Ergebnis-Set aber andere Syntax & Semantik	1
differing row contents	1
differing row contents	1
differing row contents	1
zufällig gleiches Ergebnis-Set aber andere Syntax & Semantik → Test für Verfahren mit pragmatischen Ansatz	1
differing row contents	1
differing row contents	1
differing row contents	0
zufällig gleiches Ergebnis-Set aber andere Syntax & Semantik → Test für Verfahren mit pragmatischen Ansatz	1

	1
	0
	0
	0
	0
	0
differing row contents	1
differing row contents	1
differing row contents	1
differing row contents	0
differing row contents	1
differing row contents	0
differing row contents	1
differing row contents	1
	1
differing row contents	0

	CS12x
differing row contents	1
differing row contents	1
Kandidat-query falsch → mysql spez. Problem – old query (fehlerhaft mit default mysql settings): `SELECT InvoiceId, CustomerId, count(CustomerId) FROM Invoice;`	0
	0
differing row contents	1
differing row contents	1
differing row contents	0
	1

	0
	0
	0
WrongNumColumnsError [-0]: More or fewer columns included.	0

		%
Expected != actual:	28	49,1
error	0	0