

CLAIMS

What is claimed is:

1. A data query method, wherein the method is executed by a database system, the database system runs on a computer device, and the method comprises:

5 receiving a data query request;

generating an original query plan according to the data query request, wherein the original query plan is a query plan generated by parsing the data query request for obtaining a data in a database;

10 restructuring a join predicate in the original query plan, wherein the join predicate is a predicate used to describe a relationship of columns in two or more tables in the database and used to combine two or more tables in the database into one result table;

determining basic information of a restructured join predicate in the original query plan;

15 determining, based on a join operation rule, a constraint condition of the restructured join predicate in the original query plan, wherein the constraint condition is a condition added to the restructured join predicate in the original query plan if the join operation rule is not satisfied;

20 determining an equal-cost query plan for the original query plan according to the basic information of the restructured join predicate in the original query plan and the constraint condition of the restructured join predicate in the original query plan, wherein the equal-cost query plan is a query plan obtained from the original query plan by means of transformation by using operational-level rules or predicate-level rules; and

performing querying according to the equal-cost query plan for the original query plan.

25 2. The method according to claim 1, wherein the determining an equal-cost query plan for the original query plan according to the basic information of the restructured join predicate in the original query plan and the constraint condition of the restructured join predicate in the original query plan comprises:

determining an equal-cost query plan set for the original query plan according to the basic information of the restructured join predicate in the original query plan and the constraint condition of the restructured join predicate in the original query plan; and

30 the performing querying according to the equal-cost query plan for the original query plan comprises:

selecting a query plan from the equal-cost query plan set to perform querying.

3. The method according to claim 2, wherein the method further comprises:

obtaining a candidate query plan set according to the original query plan; and

the determining an equal-cost query plan set for the original query plan according to the basic information of the restructured join predicate in the original query plan and the constraint condition of the restructured join predicate in the original query plan comprises:

5 determining the equal-cost query plan set from the candidate query plan set according to the basic information of the restructured join predicate in the original query plan and the constraint condition of the restructured join predicate in the original query plan.

4. The method according to any one of claims 1 to 3, wherein the restructuring a join predicate in the original query plan comprises:

10 transforming all join predicates of an inner join operation in the original query plan into a conjunctive normal form, and combining join predicates having a same input table; and

combining all join predicates of a non-inner join operation in the original query plan into one join predicate.

5. The method according to claim 1, wherein the basic information of the join predicate
15 comprises an input table of the join predicate, an input table of the join predicate that is comprised in a left subtree of a join operation in which the join predicate is located, an input table of the join predicate that is comprised in a right subtree of the join operation in which the join predicate is located, and a join type of the join operation in which the join predicate is located.

20 6. The method according to claim 1, wherein the join operation rule comprises an operational-level rule and a predicate-level rule, the operational-level rule comprises an operational-level exchange rule: $e_1 o_{p12}^a e_2 = e_2 o_{p12}^a e_1$, an operational-level combination rule: $(e_1 o_{p12}^a e_2) o_{p23}^b e_3 = e_1 o_{p12}^a (e_2 o_{p23}^b e_3)$, an operational-level left-permutation rule: $(e_1 o_{p12}^a e_2) o_{p13}^b e_3 = (e_1 o_{p13}^b e_3) o_{p12}^a e_2$, and an operational-level right-permutation rule: $e_1 o_{p13}^a (e_2 o_{p23}^b e_3) = e_2 o_{p23}^b (e_1 o_{p13}^a e_3)$, the predicate-level rule comprises a predicate-level
25 combination rule: $(e_1 o_{p12}^a e_2) o_{p \wedge p23}^b e_3 = e_1 o_{p12 \wedge p}^a (e_2 o_{p23}^b e_3)$, a predicate-level left-permutation rule: $(e_1 o_{p12}^a e_2) o_{p13 \wedge p}^b e_3 = (e_1 o_{p13}^b e_3) o_{p12 \wedge p}^a e_2$, and a predicate-level right-permutation rule: $e_1 o_{p \wedge p13}^a (e_2 o_{p23}^b e_3) = e_2 o_{p \wedge p23}^b (e_1 o_{p13}^a e_3)$, e_i is a query expression, o^a and o^b are join operations, and p_{ij} is a join predicate between e_i and e_j .

30 7. The method according to claim 6, wherein the determining, based on a join operation rule, a constraint condition of the restructured join predicate in the original query plan comprises:

traversing, for each join operation node o^B in the original query plan, each join operation node o^A in a subtree that uses o^B as a root node; and

when o^A is in a left subtree of the subtree that uses o^B as the root node, if o^A and o^B do not satisfy the operational-level combination rule and the predicate-level combination rule, adding a first constraint condition to each restructured join predicate on o^B ; and if o^A and o^B do not satisfy the operational-level left-permutation rule and the predicate-level left-permutation rule, adding a second constraint condition to each restructured join predicate on o^B ; or

when o^A is in a right subtree of the subtree that uses o^B as the root node, if o^A and o^B do not satisfy the operational-level combination rule and the predicate-level combination rule, adding a second constraint condition to each restructured join predicate on o^B ; and if o^A and o^B do not satisfy the operational-level right-permutation rule and the predicate-level right-permutation rule, adding a first constraint condition to each restructured join predicate on o^B ; wherein

the first constraint condition is $T(e_2) \rightarrow T(e_1) \cap T(p_{a1} \wedge p_{a2} \wedge \dots \wedge p_{an})$, the second constraint condition is $T(e_1) \rightarrow T(e_2) \cap T(p_{a1} \wedge p_{a2} \wedge \dots \wedge p_{an})$, $T(e_1)$ indicates an original input table comprised in a left subtree e_1 of a subtree that uses o^A as a root node, $T(e_2)$ indicates an original input table comprised in a right subtree e_2 of the subtree that uses o^A as the root node, $T(p_{a1} \wedge p_{a2} \wedge \dots \wedge p_{an})$ indicates an original input table comprised in a restructured join predicate $p_{a1} \wedge p_{a2} \wedge \dots \wedge p_{an}$ on o^A , and $T_1 \rightarrow T_2$ indicates that if a first subtree that uses a join operation in which a join predicate constrained by $T_1 \rightarrow T_2$ is located as a root node comprises any input table in T_1 , the first subtree comprises all input tables in T_2 .

8. The method according to claim 1, wherein the determining an equal-cost query plan for the original query plan according to the basic information of the restructured join predicate in the original query plan and the constraint condition of the restructured join predicate in the original query plan comprises:

determining that a first candidate query plan P' is an equal-cost query plan for the original query plan P if P' satisfies the following conditions:

a join type of each first join operation of P' is an inner join, the first join operation is a join operation having at least two restructured join predicates, and a join type of a join operation that is in P and that is corresponding to each join predicate in the at least two

restructured join predicates is an inner join;

for each join predicate p on each join operation $o^{b'}$ in P' ,

when $o^{b'}$ does not satisfy the operational-level exchange rule, $T_l(p)$ in P' and $T_l(p)$ in P are equal, and $T_r(p)$ in P' and $T_r(p)$ in P are equal, wherein $T_l(p)$ indicates an input table of p comprised in a left subtree of a join operation in which p is located, and $T_r(p)$ indicates an input table of p comprised in a right subtree of the join operation in which p is located; or

when $o^{b'}$ satisfies the operational-level exchange rule, $T_l(p)$ in P' and $T_l(p)$ in P are equal, and $T_r(p)$ in P' and $T_r(p)$ in P are equal; or $T_l(p)$ in P' and $T_r(p)$ in P are equal, and $T_r(p)$ in P' and $T_l(p)$ in P are equal; and

for each join predicate p on each join operation $o^{b'}$ in P' , p satisfies a constraint condition for p in P .

9. The method according to claim 2, wherein the selecting a query plan from the equal-cost query plan set to perform querying comprises:

selecting a query plan with minimum costs from the equal-cost query plan set to perform querying.

10. A data query apparatus, comprising:

a receiving module, configured to receive a data query request;

a generation module, configured to generate an original query plan according to the data query request, wherein the original query plan is a query plan generated by parsing the data query request for obtaining a data in a database;

a restructuring module, configured to restructure a join predicate in the original query plan, wherein the join predicate is a predicate used to describe a relationship of columns in two or more tables in the database and used to combine two or more tables in the database into one result table;

a determining module, configured to: determine basic information of a restructured join predicate in the original query plan; determine, based on a join operation rule, a constraint condition of the restructured join predicate in the original query plan, wherein the constraint condition is a condition added to the restructured join predicate in the original query plan if the join operation rule is not satisfied; and determine an equal-cost query plan for the original query plan according to the basic information of the restructured join predicate in the original query plan and the constraint condition of the restructured join predicate in the original query plan, wherein the equal-cost query plan is a query plan obtained from the original query plan by means of transformation by using operational-level rules or predicate-level rules; and

a query module, configured to perform querying according to the equal-cost query plan

for the original query plan.

11. The apparatus according to claim 10, wherein the determining module is specifically configured to determine an equal-cost query plan set for the original query plan according to the basic information of the restructured join predicate in the original query plan and the
5 constraint condition of the restructured join predicate in the original query plan; and

the query module is specifically configured to select a query plan from the equal-cost query plan set to perform querying.

12. The apparatus according to claim 11, wherein the apparatus further comprises:

an obtaining module, configured to obtain a candidate query plan set according to the
10 original query plan; wherein

the determining module is specifically configured to determine the equal-cost query plan set from the candidate query plan set according to the basic information of the restructured join predicate in the original query plan and the constraint condition of the restructured join predicate in the original query plan.

13. The apparatus according to any one of claims 10 to 12, wherein the restructuring
15 module is specifically configured to:

transform all join predicates of an inner join operation in the original query plan into a conjunctive normal form, and combine join predicates having a same input table; and

combine all join predicates of a non-inner join operation in the original query plan into
20 one join predicate.

14. The apparatus according to claim 10, wherein the basic information of the join predicate comprises an input table of the join predicate, an input table of the join predicate that is comprised in a left subtree of a join operation in which the join predicate is located, an input table of the join predicate that is comprised in a right subtree of the join operation in
25 which the join predicate is located, and a join type of the join operation in which the join predicate is located.

15. The apparatus according to claim 10, wherein the join operation rule comprises an operational-level rule and a predicate-level rule, wherein the operational-level rule comprises an operational-level exchange rule: $e_1 o_{p12}^a e_2 = e_2 o_{p12}^a e_1$, an operational-level combination
30 rule: $(e_1 o_{p12}^a e_2) o_{p23}^b e_3 = e_1 o_{p12}^a (e_2 o_{p23}^b e_3)$, an operational-level left-permutation rule: $(e_1 o_{p12}^a e_2) o_{p13}^b e_3 = (e_1 o_{p13}^b e_3) o_{p12}^a e_2$, and an operational-level right-permutation rule: $e_1 o_{p13}^a (e_2 o_{p23}^b e_3) = e_2 o_{p23}^b (e_1 o_{p13}^a e_3)$, the predicate-level rule comprises a predicate-level combination rule: $(e_1 o_{p12}^a e_2) o_{p \wedge p23}^b e_3 = e_1 o_{p12 \wedge p}^a (e_2 o_{p23}^b e_3)$, a predicate-level left-

permutation rule: $(e_1 \ o^a_{p_{12}} \ e_2) \ o^b_{p_{13 \wedge p}} \ e_3 = (e_1 \ o^b_{p_{13}} \ e_3) \ o^a_{p_{12 \wedge p}} \ e_2$, and a predicate-level right-

permutation rule: $e_1 \ o^a_{p \wedge p_{13}} \ (e_2 \ o^b_{p_{23}} \ e_3) = e_2 \ o^b_{p \wedge p_{23}} \ (e_1 \ o^a_{p_{13}} \ e_3)$, e_i is a query expression, o^a and o^b are join operations, and p_{ij} is a join predicate between e_i and e_j .

16. The apparatus according to claim 15, wherein the determining module is specifically configured to:

traverse, for each join operation node o^B in the original query plan, each join operation node o^A in a subtree that uses o^B as a root node; and

when o^A is in a left subtree of the subtree that uses o^B as the root node, if o^A and o^B do not satisfy the operational-level combination rule and the predicate-level combination rule, add a first constraint condition to each restructured join predicate on o^B ; and if o^A and o^B do not satisfy the operational-level left-permutation rule and the predicate-level left-permutation rule, add a second constraint condition to each restructured join predicate on o^B ; or

when o^A is in a right subtree of the subtree that uses o^B as the root node, if o^A and o^B do not satisfy the operational-level combination rule and the predicate-level combination rule, add a second constraint condition to each restructured join predicate on o^B ; and if o^A and o^B do not satisfy the operational-level right-permutation rule and the predicate-level right-permutation rule, add a first constraint condition to each restructured join predicate on o^B ; wherein

the first constraint condition is $T(e_2) \rightarrow T(e_1) \cap T(p_{a1} \wedge p_{a2} \wedge \dots \wedge p_{an})$, the second constraint condition is $T(e_1) \rightarrow T(e_2) \cap T(p_{a1} \wedge p_{a2} \wedge \dots \wedge p_{an})$, $T(e_1)$ indicates an original input table comprised in a left subtree e_1 of a subtree that uses o^A as a root node, $T(e_2)$ indicates an original input table comprised in a right subtree e_2 of the subtree that uses o^A as the root node, $T(p_{a1} \wedge p_{a2} \wedge \dots \wedge p_{an})$ indicates an original input table comprised in a restructured join predicate $p_{a1} \wedge p_{a2} \wedge \dots \wedge p_{an}$ on o^A , and $T_1 \rightarrow T_2$ indicates that if a first subtree that uses a join operation in which a join predicate constrained by $T_1 \rightarrow T_2$ is located as a root node comprises any input table in T_1 , the first subtree comprises all input tables in T_2 .

17. The apparatus according to claim 10, wherein the determining module is specifically configured to:

determine that a first candidate query plan P' is an equal-cost query plan for the original query plan P if P' satisfies the following conditions:

a join type of each first join operation of P' is an inner join, the first join operation is a join operation having at least two restructured join predicates, and a join type of a join operation that is in P and that is corresponding to each join predicate in the at least two restructured join predicates is an inner join;

5 for each join predicate p on each join operation $o^{b'}$ in P' ,

when $o^{b'}$ does not satisfy the operational-level exchange rule, $T_l(p)$ in P' and $T_l(p)$ in P are equal, and $T_r(p)$ in P' and $T_r(p)$ in P are equal, wherein $T_l(p)$ indicates an input table of p comprised in a left subtree of a join operation in which p is located, and $T_r(p)$ indicates an input table of p comprised in a right subtree of the join operation in which p is located; or

10 when $o^{b'}$ satisfies the operational-level exchange rule, $T_l(p)$ in P' and $T_l(p)$ in P are equal, and $T_r(p)$ in P' and $T_r(p)$ in P are equal; or $T_l(p)$ in P' and $T_r(p)$ in P are equal, and $T_r(p)$ in P' and $T_l(p)$ in P are equal; and

for each join predicate p on each join operation $o^{b'}$ in P' , p satisfies a constraint condition for p in P .

15 18. The apparatus according to claim 11, wherein the query module is specifically configured to:

select a query plan with minimum costs from the equal-cost query plan set to perform querying.

20 19. A database system, comprising the data query apparatus according to claim 10 and a database.

20. A computer-readable recording medium on which a program is recorded, wherein the program, when executed, enables the computer to perform the steps of:

receiving a data query request;

25 generating an original query plan according to the data query request, wherein the original query plan is a query plan generated by parsing the data query request for obtaining a data in a database;

restructuring a join predicate in the original query plan, wherein the join predicate is a predicate used to describe a relationship of columns in two or more tables in the database and used to combine two or more tables in the database into one result table;

30 determining basic information of a restructured join predicate in the original query plan;

determining, based on a join operation rule, a constraint condition of the restructured join predicate in the original query plan, wherein the constraint condition is a condition added to the restructured join predicate in the original query plan if the join operation rule is not satisfied;

- determining an equal-cost query plan for the original query plan according to the basic information of the restructured join predicate in the original query plan and the constraint condition of the restructured join predicate in the original query plan, wherein the equal-cost query plan is a query plan obtained from the original query plan by means of transformation
- 5 by using operational-level rules or predicate-level rules; and
- performing querying according to the equal-cost query plan for the original query plan.