Otus PHP Pro, 2.3 (#11) - PostgreSQL, ( )

Analytics by queries (using indexes)

Everywhere we look at the execution time of requests and enter data into the table:

Let the icons denote:

- there is an acceleration from the index
- no acceleration
- negative acceleration (index hinders)

SQL query	Index created	Small data (~1000 rows)			Medium data (~10.000 rows)			Large data (~ 100.000 rows)		
		Exec time (without index)	Exec time (with index)	Search speedup	Exec time (without index)	Exec time (with index)	Search speedup	Exec time (without index)	Exec time (with index)	Search speedup
explain analyse select "name" from "movie" where duration >= 90;	create index movie_duration_index on movie (duration);	0.428	6.340	0,067	2,014	2,241	0,89	22.181	21.307	1,04
explain analyse select * from "order" where (paytime::date <= now()::date) and (paytime::date >= (now() - interval '7 day')::date);	create index on "order" (date (paytime));	0.175	0.068	2,57	0,843	0,409	2,06	64.716	6.296	10,2
explain analyze select count(*) from "user" where length("name") = 10;	create index i_user_name_length on "user" using btree (length ("name"));	0.632	0.199	3,17	0,397	0,395	1	31.534	2.354	13,4
explain analyse select m.name as movie_name, count(o.id) as tickets_sold, sum(schedule.price) as revenue from "order" as o left join "schedule" on "schedule".id = o.schedule_id left join "movie" m on schedule.movie_id = m.id where (o.paytime::date <= now()::date) and (o.paytime::date >= (now() - interval '30 day')::date) group by movie_name order by revenue desc;	create index i_order_paytime on "order" (date (paytime));	0.695	0.587	1,18	38,172	32,092	1,19	677.711	426.409	1,5

explain analyze select u.name from "order" as o left join "schedule" on "schedule".id = o.schedule_id left join "movie" m on schedule.movie_id = m.id left join "user" as u on o."user_id" = u.id where length(m."name") = 10;	create index i_movie_name_length on "movie" using btree (length ("name"));	0.693	0.579	1,19	8,192	5,948	1.38	89.809	75.793	1,18
explain analyze select m.name, s.price from schedule s join movie m on s.movie_id = m.id where (s.start_time::date >= (now() - interval '30 day')::date) and s.price <= 500 order by s.price desc limit 10;	create index i_schedule_time_price on "schedule" using btree (date (start_time), "price" desc);	0.719	0.743	0,96	9.236	13.457	0.68	142.467	148.566	0,96