1.List all taxpayers who have paid less total tax than what they should have paid.

```
select inc.pan_number,sum(tp.payment_amount) as paid_tax ,   sum(income_amount*tax_rate)
as total_tax from income as inc  join tax_rate on
 tax_type=income_type join tax_payments as tp on tp.pan_number=inc.pan_number
 group by inc.pan_number
 Having sum(tp.payment_amount)<sum(income_amount*tax_rate);
```

2.Select the total tax revenue for each branch for the year 2023 and total revenue in descending order:

```
SELECT itb.office_id, SUM(tpp.payment_amount) as total_revenue from  tax_payments
as tpp
join tax_payer as tp on tp.pan_number=tpp.pan_number
join income_tax_branches as  itb on tp.office_id=itb.office_id
WHERE tpp.tax_year = 2023
GROUP BY itb.office_id
ORDER BY total_revenue DESC;
```

3.Retrieve pan_number,total_income,total_tax_paid of all tax payers for the year=2022.

```
SELECT  Income.pan_number,SUM(income_amount) AS total_income,
SUM(Payment_Amount) AS total_tax_paid FROM Income
JOIN Tax_Payments ON Income.Pan_number = Tax_Payments.pan_number AND
Income.income_date BETWEEN '2022-01-01' AND '2022-12-31'
WHERE  Tax_Payments.Tax_year = 2022
group by Income.pan_number;
```

4.Get a list of taxpayers who have been audited in the last two years, and show the total amount of taxes paid by them and the number of times they have been audited:

```
SELECT tax_payer.pan_number, tax_payer."First_name",tax_payer."Last_name",
COUNT(tax_audits.audit_id) as audit_count, SUM(tax_payments.payment_amount) as
total_tax_paid
```

```
FROM tax_payer
JOIN tax_audits ON tax_payer.pan_number = tax_audits.pan_number
JOIN tax_payments ON tax_payer.pan_number = tax_payments.pan_number
WHERE (tax_payments.tax_year = '2022' or tax_payments.tax_year = '2023')
GROUP BY tax_payer.pan_number;
```

5.List pan_number,first_name,last_name,email_id of taxpayers who have not paid tax this year.

```
SELECT tp.pan_number,tp."First_name",tp."Last_name",ep.email_id
FROM tax_payer as tp
LEFT JOIN tax_payments ON tp.pan_number = tax_payments.pan_number AND
tax_payments.tax_year = '2023'
join email_tax_payer as ep on tp.pan_number=ep.pan_number
WHERE tax_payments.payment_id IS NULL;
```

6.Give details of salaried tax_payers whose withheld tax are not paid

```
select * from tax_payer natural join
(select pan_number from income where income.income_type='salary' except
select pan_number from tax_withholdings)  as r1;
```

7.List top 10 taxpayers having a maximum number of assets.

```
select pan_number,count(asset_id) as total_assets from assets
group by pan_number
ORDER BY total_assets DESC limit 10;
```

8.List income tax branches office ids and count of questions that have not been answered by that branch in descending order.

```
select inc.office_id,count(tp.pan_number) as que_unanswered from
 income_tax_branches as inc  join
 tax_payer as tp on inc.office_id=tp.office_id join
tax_questions_asked as tqa on tqa.pan_number=tp.pan_number join
tax_questions as tq on tq.question_id=tqa.question_id where tq.status='Not answered'
group by inc.office_id
order by que_unanswered desc;
```

9. List details of tax_payers who are eligible for child tax credit but have not get credit.

```
select * from tax_payer natural join(
select pan_number from tax_dependents where relation='Child' except
select pan_number from tax_credits where credit_type='CTC' ) as r1;
```

10.List taxpayers who have not submitted the tax form after its due date.

```
select * from tax_forms where form_status='PENDING' AND form_due_date<= '2023-03-01';
```

11.List details of top tax preparers who have prepared and assessed the maximum number of taxpayers.

```
select tp.preparer_id,count(pan_number) from tax_preparers as tp join
prepares as pp on pp.preparer_id=tp.preparer_id
group by tp.preparer_id
order by count(pan_number) desc limit 5;
```

12.List details of taxpayers who have got total refund above specified limit for given year.

```
select * from tax_payer as tp join
(select pan_number,sum(refund_amount) as total_refund,tr.tax_year  from tax_refunds as tr

group by pan_number,tax_year
order by total_refund desc) as r2 on r2.pan_number=tp.pan_number
where r2.total_refund>=1000000;
```

13.Find the bank details of taxpayers having maximum number of accounts in banks.

```
select pan_number,count(account_no) from bank natural join has_account

 group by pan_number

 order by count(account_no) desc;
```